

(12)特許協力条約に基づいて公開された国際出願

(19)世界知的所有権機関
国際事務局(43)国際公開日
2001年11月1日 (01.11.2001)

PCT

(10)国際公開番号
WO 01/82611 A1(51)国際特許分類⁷:

H04N 5/93, G11B 20/10

(72)発明者; および

(75)発明者/出願人(米国についてのみ): 加藤元樹 (KATO, Motoki) [JP/JP]; 浜田俊也 (HAMADA, Toshiya) [JP/JP]; 〒141-0001 東京都品川区北品川6丁目7番35号 ソニー株式会社内 Tokyo (JP).

(21)国際出願番号:

PCT/JP01/03418

(22)国際出願日:

2001年4月20日 (20.04.2001)

(74)代理人: 小池 晃, 外 (KOIKE, Akira et al.); 〒105-0001 東京都港区虎ノ門二丁目6番4号 第11森ビル Tokyo (JP).

(25)国際出願の言語:

日本語

(81)指定国(国内): CA, US.

(26)国際公開の言語:

日本語

添付公開書類:
— 国際調査報告書

(30)優先権データ:

特願2000-121856 2000年4月21日 (21.04.2000) JP
特願2000-271551 2000年9月7日 (07.09.2000) JP

(71)出願人(米国を除く全ての指定国について): ソニー株式会社 (SONY CORPORATION) [JP/JP]; 〒141-0001 東京都品川区北品川6丁目7番35号 Tokyo (JP).

2文字コード及び他の略語については、定期発行される各PCTガゼットの巻頭に掲載されている「コードと略語のガイドスノート」を参照。

(54)Title: INFORMATION PROCESSING APPARATUS AND METHOD, RECORDED MEDIUM, AND PROGRAM

(54)発明の名称: 情報処理装置及び方法、記録媒体、並びにプログラム

a シントラス	b バイト数	c 略号
info_dvr()		
TableOfPlayLists_Start_address	32	uimsbf
MakersPrivateData_Start_address	32	uimsbf
reserved	192	bslbf
DVRVolume()		
for (i=0;i<N;i++)		
padding_word	16	bslbf
TableOfPlayLists()		
for (i=0;i<N;i++)		
padding_word	16	bslbf
MakersPrivateData()		

a...SYNTAX
b...NUMBER OF BYTES
c...SYMBOL

(57)Abstract: Information concerning all the information recorded on a disk is described in a DVRVolum(). Therefore by reading the description, the information recorded on the disk can be confirmed. Information used when information recorded on the disk is reproduced is described in a TableOfPlaylists(). By reading the description, it is possible to readily search for desired information. Consequently confirmation of the information recorded on a disk or desired information can be readily performed.

(57)要約:

WO 01/82611 A1

DVRVolum()には、ディスク内に記録されている全ての情報に関する情報が記述されている。この記述により、ディスク内に記録されている情報の確認を行うことができる。TableOfPlaylists()には、ディスク内に記録されている情報を再生する際の情報が記述されている。この記述を用いることにより、所望の情報の検索を簡便に検索することができる。これにより、ディスク内に記録されている情報の確認や所望の情報の確認を簡便に行うことができる。

明細書

情報処理装置及び方法、記録媒体、並びにプログラム

技術分野

本発明は、情報処理装置及び方法、記録媒体、並びにプログラムに関し、特に、記録媒体に記録されているデータの内容の管理情報をファイル化して記録する情報処理装置及び方法、記録媒体、並びにプログラムに関する。

背景技術

近年、記録再生装置から取り外し可能なディスク型の記録媒体として、各種の光ディスクが提案されつつある。このような記録可能な光ディスクは、数ギガバイトの大容量メディアとして提案されており、ビデオ信号等のAV (Audio Visual) 信号を記録するメディアとしての期待が高い。この記録可能な光ディスクに記録するデジタルのAV信号のソース（供給源）としては、CSデジタル衛星放送やBSデジタル放送があり、また、将来はデジタル方式の地上波テレビジョン放送等も提案されている。

ここで、これらのソースから供給されるデジタルビデオ信号は、通常MPEG (Moving Picture Experts Group) 2方式で画像圧縮されているのが一般的である。また、記録装置には、その装置固有の記録レートが定められている。従来の民生用映像蓄積メディアで、デジタル放送由来のデジタルビデオ信号を記録する場合、アナログ記録方式であれば、デジタルビデオ信号をデコード後、帯域制限をして記録する。或いは、MPEG1 Video、MPEG2 Video、DV方式をはじめとするデジタル記録方式であれば、1度デコードされた後に、その装置固有の記録レート・符号化方式で再エンコードされて記録される。

しかしながら、このような記録方法は、供給されたビットストリームを1度デ

コードし、その後で帯域制限や再エンコードを行って記録するため、画質の劣化を伴う。画像圧縮されたディジタル信号の記録をする場合、入力されたディジタル信号の伝送レートが記録再生装置の記録レートを超えない場合には、供給されたビットストリームをデコードや再エンコードすることなく、そのまま記録する方法が最も画質の劣化が少ない。但し、画像圧縮されたディジタル信号の伝送レートが記録媒体としてのディスクの記録レートを超える場合には、記録再生装置でデコード後、伝送レートがディスクの記録レートの上限以下になるように、再エンコードをして記録する必要はある。

また、入力ディジタル信号のビットレートが時間により増減する可変レート方式によって伝送されている場合には、回転ヘッドが固定回転数であるために記録レートが固定レートになるテープ記録方式に比べ、一度バッファにデータを蓄積し、バースト的に記録ができるディスク記録装置が記録媒体の容量をより無駄なく利用できる。

以上のように、ディジタル放送が主流となる将来においては、データストリーマのように放送信号をディジタル信号のまま、デコードや再エンコードすることなく記録し、記録媒体としてディスクを使用した記録再生装置が求められると予測される。

上述したように、記録媒体の容量が増大することにより、その記録媒体には、多くのデータ（この場合、番組に関する映像や音声等）が記録できるようになる。したがって、1枚のディスクに多くの番組が記録されることになり、ユーザが、それらのディスク内に記録されている多くの番組から視聴したい1番組を選択するといったような操作が煩雑になってしまふ。そこで、ユーザがディスクの再生時に、簡便に記録されているデータを確認し、所望の番組（データ）が選択できるようにする必要があるといった課題があった。

発明の開示

本発明はこのような状況に鑑みてなされたものであり、記録媒体に記録されているデータの内容の管理情報をファイル化して記録することにより、記録媒体に

記録されているデータ内容、及び、再生情報を適切に管理することができるようすることを目的とする。

本発明に係る情報処理装置は、記録媒体に記録されている情報の再生手順を指定する再生指定情報と、再生指定情報を管理する管理情報を生成する生成手段と、生成手段により生成された再生指定情報と管理情報を記録媒体に記録する記録手段とを有し、管理情報は、再生指定情報に基づく再生が終了された時点で再生指定情報に付けられていた名称に関する名称情報を含み、再生指定情報は、再生指定情報に基づく再生が終了された時点の時間情報を含む。

本発明に係る情報処理方法は、記録媒体に記録されている情報の再生手順を指定する再生指定情報と、再生指定情報を管理する管理情報を生成する生成ステップと、生成ステップの処理により生成された再生指定情報と管理情報を記録媒体に記録する記録ステップとを含み、管理情報は、再生指定情報に基づく再生が終了された時点で再生指定情報に付けられていた名称に関する名称情報を含み、再生指定情報は、再生指定情報に基づく再生が終了された時点の時間情報を含む。

本発明に係る記録媒体のプログラムは、管理情報が、再生指定情報に基づく再生が終了された時点で再生指定情報に付けられていた名称に関する名称情報を含み、再生指定情報は、再生指定情報に基づく再生が終了された時点の時間情報を含む。

本発明に係るプログラムは、管理情報が、再生指定情報に基づく再生が終了された時点で再生指定情報に付けられていた名称に関する名称情報を含み、再生指定情報は、再生指定情報に基づく再生が終了された時点の時間情報を含む。

本発明に係る情報処理装置は、再生指定情報に基づく再生が終了された時点で再生指定情報に付けられていた名称に関する名称情報を含む管理情報と、再生指定情報に基づく再生が終了された時点の時間情報を含む再生指定情報に基づいて、記録媒体の主情報の再生を制御する制御手段を備える。

本発明に係る情報処理方法は、再生指定情報に基づく再生が終了された時点で再生指定情報に付けられていた名称に関する名称情報を含む管理情報と、再生指定情報に基づく再生が終了された時点の時間情報を含む再生指定情報に基づいて、記録媒体の主情報の再生を制御する制御ステップを含む。

本発明に係る記録媒体のプログラムは、再生指定情報に基づく再生が終了された時点で再生指定情報に付けられていた名称に関する名称情報を含む管理情報と、再生指定情報に基づく再生が終了された時点の時間情報を含む再生指定情報に基づいて、記録媒体の主情報の再生を制御する制御ステップを含む。

本発明に係るプログラムは、記録されている主情報の再生手順を指定する再生指定情報と、再生指定情報を管理する管理情報を記録されている記録媒体から、主情報を再生する情報処理装置を制御するコンピュータに、再生指定情報に基づく再生が終了された時点で再生指定情報に付けられていた名称に関する名称情報を含む管理情報と、再生指定情報に基づく再生が終了された時点の時間情報を含む再生指定情報に基づいて、記録媒体の主情報の再生を制御する制御ステップを実行させる。

本発明に係る記録媒体は、管理情報が、再生指定情報に基づく再生が終了された時点で再生指定情報に付けられていた名称に関する名称情報を含み、再生指定情報は、再生指定情報に基づく再生が終了された時点の時間情報を含む。

本発明に係る情報処理装置は、記録媒体に記録されている情報の再生手順を指定する再生指定情報と、再生指定情報を管理する管理情報を生成する生成手段と、生成手段により生成された再生指定情報と管理情報を記録媒体に記録する記録手段とを有し、管理情報は、管理情報が管理する全ての再生指定情報についての閲覧の許可に関する閲覧許可情報を含み、再生指定情報は、再生指定情報の閲覧の許可に関する閲覧許可情報を含む。

本発明に係る情報処理方法は、記録媒体に記録されている情報の再生手順を指定する再生指定情報と、再生指定情報を管理する管理情報を生成する生成ステップと、生成ステップの処理により生成された再生指定情報と管理情報を記録媒体に記録する記録ステップとを含み、管理情報は、管理情報が管理する全ての再生指定情報についての閲覧の許可に関する閲覧許可情報を含み、再生指定情報は、再生指定情報の閲覧の許可に関する閲覧許可情報を含む。

本発明に係る記録媒体のプログラムは、管理情報が、管理情報が管理する全ての再生指定情報についての閲覧の許可に関する閲覧許可情報を含み、再生指定情報は、再生指定情報の閲覧の許可に関する閲覧許可情報を含む。

本発明に係るプログラムは、管理情報が、管理情報が管理する全ての再生指定情報についての閲覧の許可に関する閲覧許可情報を含み、再生指定情報は、再生指定情報の閲覧の許可に関する閲覧許可情報を含む。

本発明に係る記録媒体は、管理情報が、管理情報が管理する全ての再生指定情報についての閲覧の許可に関する閲覧許可情報を含み、再生指定情報は、再生指定情報の閲覧の許可に関する閲覧許可情報を含む。

本発明に係る情報処理装置は、記録媒体に記録されている情報の再生手順を指定する再生指定情報と、再生指定情報を管理する管理情報を生成する生成手段と、生成手段により生成された管理情報を記録媒体に記録する記録手段とを有し、管理情報は、管理情報が管理する全ての再生指定情報を再生順に登録する再生順序情報を含み、再生指定情報は、その再生区間の時間情報を含む。

本発明に係る情報処理方法は、記録媒体に記録されている情報の再生手順を指定する再生指定情報と、再生指定情報を管理する管理情報を生成する生成ステップと、生成ステップの処理により生成された管理情報を記録媒体に記録する記録ステップとを含み、管理情報は、管理情報が管理する全ての再生指定情報を再生順に登録する再生順序情報を含み、再生指定情報は、その再生区間の時間情報を含む。

本発明に係る記録媒体のプログラムは、管理情報が、管理情報が管理する全ての再生指定情報を再生順に登録する再生順序情報を含み、再生指定情報は、その再生区間の時間情報を含む。

本発明に係るプログラムは、管理情報が、管理情報が管理する全ての再生指定情報を再生順に登録する再生順序情報を含み、再生指定情報は、その再生区間の時間情報を含む。

本発明に係る情報処理装置は、管理情報が管理する全ての再生指定情報を再生順に登録する再生順序情報を含む前記管理情報と、再生区間の時間情報を含む再生指定情報に基づいて、記録媒体の主情報の再生を制御する制御手段を備える。

本発明に係る情報処理方法は、管理情報が管理する全ての再生指定情報を再生順に登録する再生順序情報を含む管理情報と、再生区間の時間情報を含む再生指定情報に基づいて、記録媒体の主情報の再生を制御する制御ステップを含む。

本発明に係る記録媒体のプログラムは、管理情報が管理する全ての再生指定情報を再生順に登録する再生順序情報を含む管理情報と、再生区間の時間情報を含む再生指定情報に基づいて、記録媒体の主情報の再生を制御する制御ステップを含む。

本発明に係るプログラムは、記録されている主情報の再生手順を指定する再生指定情報と、再生指定情報を管理する管理情報が記録されている記録媒体から、主情報を再生する情報処理装置を制御するコンピュータに、管理情報が管理する全ての再生指定情報を再生順に登録する再生順序情報を含む管理情報と、再生区間の時間情報を含む再生指定情報に基づいて、記録媒体の主情報の再生を制御する制御ステップを実行させる。

本発明に係る記録媒体は、管理情報が、管理情報が管理する全ての再生指定情報を再生順に登録する再生順序情報を含み、再生指定情報は、その再生区間の時間情報を含む。

本発明に係る情報処理装置及び方法、記録媒体、並びにプログラム、並びに第2の情報処理装置及び方法、記録媒体、並びにプログラムにおいては、管理情報は、再生指定情報に基づく再生が終了された時点で再生指定情報に付けられていた名称に関する名称情報を含み、再生指定情報は、再生指定情報に基づく再生が終了された時点の時間情報を含む。

本発明に係る情報処理装置及び方法、記録媒体、並びにプログラム、並びに第4の情報処理装置及び方法、記録媒体、並びにプログラムにおいては、管理情報は、管理情報が管理する全ての再生指定情報についての閲覧の許可に関する閲覧許可情報を含み、再生指定情報は、再生指定情報の閲覧の許可に関する閲覧許可情報を含む。

本発明に係る情報処理装置及び方法、記録媒体、並びにプログラムにおいては、管理情報は、管理情報が管理する全ての再生指定情報を再生順に登録する再生順序情報を含み、再生指定情報は、その再生区間の時間情報を含む。

本発明の更に他の目的、特徴や利点は、後述する本発明の実施例や添付する図面に基づくより詳細な説明によって明らかになるであろう。

図面の簡単な説明

図1は、本発明を適用した記録再生装置の構成を示す図である。

図2は、記録再生装置により記録媒体に記録されるデータのフォーマットについて説明する図である。

図3は、Real PlayListとVirtual PlayListについて説明する図である。

図4A～図4Cは、Real PlayListの作成について説明する図である。

図5A～図5Cは、Real PlayListの削除について説明する図である。

図6A及び図Bは、アセンブル編集について説明する図である。

図7は、Virtual PlayListにサブパスを設ける場合について説明する図である。

図8は、PlayListの再生順序の変更について説明する図である。

図9は、PlayList上のマークとClip上のマークについて説明する図である。

図10は、メニュー サムネイルについて説明する図である。

図11は、PlayListに付加されるマークについて説明する図である。

図12は、クリップに付加されるマークについて説明する図である。

図13は、PlayList、Clip、サムネイルファイルの関係について説明する図である。

図14は、ディレクトリ構造について説明する図である。

図15は、info.dvrのシンタクスを示す図である。

図16は、DVR volumeのシンタクスを示す図である。

図17は、Resumevolumeのシンタクスを示す図である。

図18は、UIAppInfovolumeのシンタクスを示す図である。

図19は、Character set valueのテーブルを示す図である。

図20は、TableOfPlayListのシンタクスを示す図である。

図21は、TableOfPlayListの他のシンタクスを示す図である。

図22は、MakersPrivateDataのシンタクスを示す図である。

図23は、xxxxx.rplsとyyyyy.vplsのシンタクスを示す図である。

図24A～図24Cは、PlayListについて説明する図である。

図25は、PlayListのシンタクスを示す図である。

- 図26は、PlayList_typeのテーブルを示す図である。
- 図27は、UIAppinfoPlayListのシンタクスを示す図である。
- 図28A～図28Cは、図27に示したUIAppinfoPlayListのシンタクス内のフラグについて説明する図である。
- 図29は、PlayItemについて説明する図である。
- 図30は、PlayItemについて説明する図である。
- 図31は、PlayItemについて説明する図である。
- 図32は、PlayItemのシンタクスを示す図である。
- 図33は、IN_timeについて説明する図である。
- 図34は、OUT_timeについて説明する図である。
- 図35は、Connection_Conditionのテーブルを示す図である。
- 図36A～図36Dは、Connection_Conditionについて説明する図である。
- 図37は、BridgeSequenceInfoを説明する図である。
- 図38は、BridgeSequenceInfoのシンタクスを示す図である。
- 図39は、SubPlayItemについて説明する図である。
- 図40は、SubPlayItemのシンタクスを示す図である。
- 図41は、SubPath_typeのテーブルを示す図である。
- 図42は、PlayListMarkのシンタクスを示す図である。
- 図43は、Mark_typeのテーブルを示す図である。
- 図44は、Mark_time_stampを説明する図である。
- 図45は、zzzzz.clipのシンタクスを示す図である。
- 図46は、ClipInfoのシンタクスを示す図である。
- 図47は、Clip_stream_typeのテーブルを示す図である。
- 図48は、offset_SPNについて説明する図である。
- 図49は、offset_SPNについて説明する図である。
- 図50A及び図50Bは、STC区間について説明する図である。
- 図51は、STC_Infoについて説明する図である。
- 図52は、STC_Infoのシンタクスを示す図である。
- 図53は、ProgramInfoを説明する図である。

図 5 4 は、ProgramInfoのシンタクスを示す図である。

図 5 5 は、VideoCodingInfoのシンタクスを示す図である。

図 5 6 は、Video_formatのテーブルを示す図である。

図 5 7 は、frame_rateのテーブルを示す図である。

図 5 8 は、display_aspect_ratioのテーブルを示す図である。

図 5 9 は、AudioCodingInfoのシンタクスを示す図である。

図 6 0 は、audio_codingのテーブルを示す図である。

図 6 1 は、audio_component_typeのテーブルを示す図である。

図 6 2 は、sampling_frequencyのテーブルを示す図である。

図 6 3 は、CPIについて説明する図である。

図 6 4 は、CPIについて説明する図である。

図 6 5 は、CPIのシンタクスを示す図である。

図 6 6 は、CPI_typeのテーブルを示す図である。

図 6 7 は、ビデオEP_mapについて説明する図である。

図 6 8 は、EP_mapについて説明する図である。

図 6 9 は、EP_mapについて説明する図である。

図 7 0 は、EP_mapのシンタクスを示す図である。

図 7 1 は、EP_type valuesのテーブルを示す図である。

図 7 2 は、EP_map_for_one_stream_PIDのシンタクスを示す図である。

図 7 3 は、TU_mapについて説明する図である。

図 7 4 は、TU_mapのシンタクスを示す図である。

図 7 5 は、ClipMarkのシンタクスを示す図である。

図 7 6 は、mark_typeのテーブルを示す図である。

図 7 7 は、mark_type_stampのテーブルを示す図である。

図 7 8 は、menu.thmbとmark.thmbのシンタクスを示す図である。

図 7 9 は、Thumbnailのシンタクスを示す図である。

図 8 0 は、thumbnail_picture_formatのテーブルを示す図である。

図 8 1 A 及び図 8 1 B は、tn_blockについて説明する図である。

図 8 2 は、DVR MPEG 2 のトランスポートストリームの構造について説明する図

である。

図 8 3 は、DVR MPEG 2 のトランスポートストリームのレコーダモデルを示す図である。

図 8 4 は、DVR MPEG 2 のトランスポートストリームのプレーヤモデルを示す図である。

図 8 5 は、source packet のシンタクスを示す図である。

図 8 6 は、TP_extra_header のシンタクスを示す図である。

図 8 7 は、copy permission indicator のテーブルを示す図である。

図 8 8 は、シームレス接続について説明する図である。

図 8 9 は、シームレス接続について説明する図である。

図 9 0 は、シームレス接続について説明する図である

図 9 1 は、シームレス接続について説明する図である。

図 9 2 は、シームレス接続について説明する図である

図 9 3 は、オーディオのオーバーラップについて説明する図である。

図 9 4 は、BridgeSequence を用いたシームレス接続について説明する図である。

図 9 5 は、BridgeSequence を用いないシームレス接続について説明する図である。

図 9 6 は、DVR STD モデルを示す図である。

図 9 7 は、復号、表示のタイミングチャートを示す図である。

図 9 8 は、info.dvr の作成／更新の処理を説明するフローチャートである。

図 9 9 は、プレイリストを再生する処理を説明するフローチャートである。

図 1 0 0 は、PlayList の再生順序を変更する処理を説明するフローチャートである。

図 1 0 1 は、媒体を説明する図である。

発明を実施するための最良の形態

以下に、本発明が適用された情報処理装置及び方法、記録媒体、並びにプログラムについて、図面を参照して説明する。図 1 は、本発明を適用した記録再生装

置 1 の内部構成例を示す図である。先ず、外部から入力された信号を記録媒体に記録する動作を行う部分の構成について説明する。記録再生装置 1 は、アナログデータ、又は、デジタルデータを入力し、記録することができる。

端子 1 1 には、アナログのビデオ信号が、端子 1 2 には、アナログのオーディオ信号が、それぞれ入力される。端子 1 1 に入力されたビデオ信号は、解析部 1 4 と A V エンコーダ 1 5 に、それぞれ出力される。端子 1 2 に入力されたオーディオ信号は、A V エンコーダ 1 5 に出力される。解析部 1 4 は、入力されたビデオ信号からシーンチェンジ等の特徴点を抽出する。

A V エンコーダ 1 5 は、入力されたビデオ信号とオーディオ信号を、それぞれ符号化し、符号化ビデオストリーム (V) 、符号化オーディオストリーム (A) 、及び A V 同期等のシステム情報 (S) をマルチブレクサ 1 6 に出力する。

符号化ビデオストリームは、例えば、M P E G (Moving Picture Expert Group) 2 方式により符号化されたビデオストリームであり、符号化オーディオストリームは、例えば、M P E G 1 方式により符号化されたオーディオストリームや、ドルビー A C 3 方式により符号化されたオーディオストリーム等である。マルチブレクサ 1 6 は、入力されたビデオ及びオーディオのストリームを、入力システム情報に基づいて多重化して、スイッチ 1 7 を介して多重化ストリーム解析部 1 8 とソースパケットタイザ 1 9 に出力する。

多重化ストリームは、例えば、M P E G 2 トランスポートストリームや M P E G 2 プログラムストリームである。ソースパケットタイザ 1 9 は、入力された多重化ストリームを、そのストリームを記録させる記録媒体 1 0 0 のアプリケーションフォーマットに従って、ソースパケットから構成される A V ストリームを符号化する。A V ストリームは、E C C (誤り訂正) 符号化部 2 0 、変調部 2 1 で所定の処理が施され、書込部 2 2 に出力される。書込部 2 2 は、制御部 2 3 から出力される制御信号に基づいて、記録媒体 1 0 0 に A V ストリームファイルを書き込む (記録する) 。

デジタルインタフェース又はデジタルテレビジョンチューナから入力されるデジタルテレビジョン放送等のトランスポートストリームは、端子 1 3 に入力される。端子 1 3 に入力されたトランスポートストリームの記録方式には、2

通りあり、それらは、トランスペアレントに記録する方式と、記録ビットレートを下げる等の目的のために再エンコードをした後に記録する方式である。記録方式の指示情報は、ユーザインターフェースとしての端子 24 から制御部 23 へ入力される。

入力トランスポートストリームをトランスペアレントに記録する場合、端子 13 に入力されたトランスポートストリームは、多重化ストリーム解析部 18 と、ソースパケットタイザ 19 に出力される。これ以降の記録媒体 100 へ AVストリームが記録されるまでの処理は、上述の入力オーディオ浸透とビデオ信号を符号化して記録する場合と同一の処理なので、その説明は省略する。

入力トランスポートストリームを再エンコードした後に記録する場合、端子 13 に入力されたトランスポートストリームは、デマルチブレクサ 26 に入力される。デマルチブレクサ 26 は、入力されたトランスポートストリームに対してデマルチブレクス処理を施し、ビデオストリーム (V) 、オーディオストリーム (A) 、及びシステム情報 (S) を抽出する。

デマルチブレクサ 26 により抽出されたストリーム (情報) の内、ビデオストリームは AV デコーダ 27 に、オーディオストリームとシステム情報はマルチブレクサ 16 に、それぞれ出力される。AV デコーダ 27 は、入力されたビデオストリームを復号し、その再生ビデオ信号を AV エンコーダ 15 に出力する。AV エンコーダ 15 は、入力ビデオ信号を符号化し、符号化ビデオストリーム (V) をマルチブレクサ 16 に出力する。

一方、デマルチブレクサ 26 から出力され、マルチブレクサ 16 に入力されたオーディオストリームとシステム情報、及び、AV エンコーダ 15 から出力されたビデオストリームは、入力システム情報に基づいて、多重化されて、多重化ストリームとして多重化ストリーム解析部 18 とソースパケットタイザ 19 にスイッチ 17 を介して出力される。これ以後の記録媒体 100 へ AVストリームが記録されるまでの処理は、上述の入力オーディオ信号とビデオ信号を符号化して記録する場合と同一の処理なので、その説明は省略する。

本例の記録再生装置 1 は、AVストリームのファイルを記録媒体 100 に記録すると共に、そのファイルを説明するアプリケーションデータベース情報も記録

する。アプリケーションデータベース情報は、制御部23により作成される。制御部23への入力情報は、解析部14からの動画像の特徴情報、多重化ストリーム解析部18からのAVストリームの特徴情報、及び端子24から入力されるユーザからの指示情報である。

解析部14から供給される動画像の特徴情報は、入力動画像信号の中の特徴的な画像に関する情報であり、例えば、プログラムの開始点、シーンチェンジ点、コマーシャル(CM)の開始・終了点等の指定情報(マーク)であり、また、その指定場所の画像のサムネイル画像の情報も含まれる。

多重化ストリーム解析部18からのAVストリームの特徴情報は、記録されるAVストリームの符号化情報に関する情報であり、例えば、AVストリーム内のIピクチャのアドレス情報、AVストリームの符号化パラメータ、AVストリームの中の符号化パラメータの変化点情報、ビデオストリームの中の特徴的な画像に関する情報(マーク)等である。

端子24からのユーザの指示情報は、AVストリームの中の、ユーザが指定した再生区間の指定情報、その再生区間の内容を説明するキャラクター文字、ユーザが好みのシーンにセットするブックマークやリリューム点の情報等である。

制御部23は、上記の入力情報に基づいて、AVストリームのデータベース(Clip)、AVストリームの再生区間(PlayItem)をグループ化したもの(PlayList)のデータベース、記録媒体100の記録内容の管理情報(info.dvr)、及びサムネイル画像の情報を生成する。これらの情報から構成されるアプリケーションデータベース情報は、AVストリームと同様にして、ECC符号化部20、変調部21で処理されて、書込部22へ入力される。書込部22は、制御部23から出力される制御信号に基づいて、記録媒体100へデータベースファイルを記録する。

上述したアプリケーションデータベース情報についての詳細は後述する。

このようにして記録媒体100に記録されたAVストリームファイル(画像データと音声データのファイル)と、アプリケーションデータベース情報が再生される場合、先ず、制御部23は、読出部28に対して、記録媒体100からアプリケーションデータベース情報を読み出すように指示する。そして、読出部28

は、記録媒体 100 からアプリケーションデータベース情報を読み出し、そのアプリケーションデータベース情報は、復調部 29、ECC復号部 30 の処理を経て、制御部 23 へ入力される。

制御部 23 は、アプリケーションデータベース情報に基づいて、記録媒体 100 に記録されている PlayList の一覧を端子 24 のユーザインタフェースへ出力する。ユーザは、PlayList の一覧から再生したい PlayList を選択し、再生を指定された PlayList に関する情報が制御部 23 へ入力される。制御部 23 は、その PlayList の再生に必要な AV ストリームファイルの読み出しを、読出部 28 に指示する。読出部 28 は、その指示に従い、記録媒体 100 から対応する AV ストリームを読み出し復調部 29 に出力する。復調部 29 に入力された AV ストリームは、所定の処理が施されることにより復調され、更に ECC復号部 30 の処理を経て、ソースデパケッタイザ 31 出力される。

ソースデパケッタイザ 31 は、記録媒体 100 から読み出され、所定の処理が施されたアプリケーションフォーマットの AV ストリームを、デマルチブレクサ 26 に出力できるストリームに変換する。デマルチブレクサ 26 は、制御部 23 により指定された AV ストリームの再生区間 (PlayItem) を構成するビデオストリーム (V) 、オーディオストリーム (A) 、及び AV 同期等のシステム情報 (S) を、AV デコーダ 27 に出力する。AV デコーダ 27 は、ビデオストリームとオーディオストリームを復号し、再生ビデオ信号と再生オーディオ信号を、それぞれ対応する端子 32 と端子 33 から出力する。

また、ユーザインタフェースとしての端子 24 から、ランダムアクセス再生や特殊再生を指示する情報が入力された場合、制御部 23 は、AV ストリームのデータベース (Clip) の内容に基づいて、記憶媒体 100 からの AV ストリームの読み出し位置を決定し、その AV ストリームの読み出しを、読出部 28 に指示する。例えば、ユーザにより選択された PlayList を、所定の時刻から再生する場合、制御部 23 は、指定された時刻に最も近いタイムスタンプを持つ I ピクチャからのデータを読み出すように読出部 28 に指示する。

また、ユーザによって高速再生 (Fast-forward playback) が指示された場合、制御部 23 は、AV ストリームのデータベース (Clip) に基づいて、AV ストリ

ームの中の I ピクチャデータを順次連続して読み出すように読み出部 28 に指示する。

読み出部 28 は、指定されたランダムアクセスポイントから AVストリームのデータを読み出し、読み出されたデータは、後段の各部の処理を経て再生される。

次に、ユーザが、記録媒体 100 に記録されている AVストリームの編集をする場合を説明する。ユーザが、記録媒体 100 に記録されている AVストリームの再生区間を指定して新しい再生経路を作成したい場合、例えば、番組 A という歌番組から歌手 A の部分を再生し、その後続けて、番組 B という歌番組の歌手 A の部分を再生したいといった再生経路を作成したい場合、ユーザインタフェースとしての端子 24 から再生区間の開始点（イン点）と終了点（アウト点）の情報が制御部 23 に入力される。制御部 23 は、AVストリームの再生区間（PlayItem）をグループ化したもの（PlayList）のデータベースを作成する。

ユーザが、記録媒体 100 に記録されている AVストリームの一部を消去したい場合、ユーザインタフェースとしての端子 24 から消去区間のイン点とアウト点の情報が制御部 23 に入力される。制御部 23 は、必要な AVストリーム部分だけを参照するように PlayList のデータベースを変更する。また、AVストリームの不必要的ストリーム部分を消去するように、書き込み部 22 に指示する。

ユーザが、記録媒体 100 に記録されている AVストリームの再生区間を指定して新しい再生経路を作成したい場合であり、且つ、それぞれの再生区間をシームレスに接続したい場合について説明する。このような場合、制御部 23 は、AVストリームの再生区間（PlayItem）をグループ化したもの（PlayList）のデータベースを作成し、更に、再生区間の接続点付近のビデオストリームの部分的な再エンコードと再多重化を行う。

先ず、端子 24 から再生区間のイン点のピクチャの情報と、アウト点のピクチャの情報が制御部 23 へ入力される。制御部 23 は、読み出部 28 にイン点側ピクチャとアウト点側のピクチャを再生するために必要なデータの読み出しを指示する。そして、読み出部 28 は、記録媒体 100 からデータを読み出し、そのデータは、復調部 29、ECC復号部 30、ソースデパケットライザ 31 を経て、デマルチブレクサ 26 に出力される。

制御部23は、デマルチブレクサ26に入力されたデータを解析して、ビデオストリームの再エンコード方法(picture_coding_typeの変更、再エンコードする符号化ビット量の割り当て)と、再多重化方式を決定し、その方式をAVエンコーダ15とマルチブレクサ16に供給する。

次に、デマルチブレクサ26は、入力されたストリームをビデオストリーム(V)、オーディオストリーム(A)、及びシステム情報(S)に分離する。ビデオストリームは、「AVデコーダ27に入力されるデータ」と「マルチブレクサ16に入力されるデータ」がある。前者のデータは、再エンコードするために必要なデータであり、これはAVデコーダ27で復号され、復号されたピクチャはAVエンコーダ15で再エンコードされて、ビデオストリームにされる。後者のデータは、再エンコードをしないで、オリジナルのストリームからコピーされるデータである。オーディオストリーム、システム情報については、直接、マルチブレクサ16に入力される。

マルチブレクサ16は、制御部23から入力された情報に基づいて、入力ストリームを多重化し、多重化ストリームを出力する。多重化ストリームは、ECC符号化部20、変調部21で処理されて、書込部22に入力される。書込部22は、制御部23から供給される制御信号に基づいて、記録媒体100にAVストリームを記録する。

以下に、アプリケーションデータベース情報や、その情報に基づく再生、編集といった操作に関する説明をする。図2は、アプリケーションフォーマットの構造を説明する図である。アプリケーションフォーマットは、AVストリームの管理のためにPlayListとClipの2つのレイヤを持つ。Volume Informationは、ディスク内の全てのClipとPlayListの管理をする。ここでは、1つのAVストリームとその付属情報のペアを1つのオブジェクトと考え、それをClipという。AVストリームファイルはClip AV stream fileといい、その付属情報は、Clip Information fileという。

1つのClip AV stream fileは、MPEG2トランスポートストリームをアプリケーションフォーマットによって規定される構造に配置したデータをストアする。一般的に、ファイルは、バイト列として扱われるが、Clip AV stream fileのコン

テンツは、時間軸上に展開され、Clipの中のエントリポイントは、主に時間ベースで指定される。所定のClipへのアクセスポイントのタイムスタンプが与えられたとき、Clip Information fileは、Clip AV stream fileの中でデータの読み出しを開始すべきアドレス情報を見つけるために役立つ。

PlayListについて、図3を参照して説明する。PlayListは、Clipの中からユーザが見たい再生区間を選択し、それを簡単に編集することができるようするために設けられている。1つのPlayListは、Clipの中の再生区間の集まりである。所定のClipの中の1つの再生区間は、PlayItemと呼ばれ、それは、時間軸上のイン点（IN）とアウト点（OUT）の対で表される。したがって、PlayListは、複数のPlayItemが集まることにより構成される。

PlayListには、2つのタイプがある。1つは、Real PlayListであり、もう1つは、Virtual PlayListである。Real PlayListは、それが参照しているClipのストリーム部分を共有している。すなわち、Real PlayListは、その参照しているClipのストリーム部分に相当するデータ容量をディスクの中で占め、Real PlayListが消去された場合、それが参照しているClipのストリーム部分もまたデータが消去される。

Virtual PlayListは、Clipのデータを共有していない。したがって、Virtual PlayListが変更又は消去されたとしても、Clipの内容には何も変化が生じない。

次に、Real PlayListの編集について説明する。図4 Aは、Real PlayListのクリエイト（create：作成）に関する図であり、AVストリームが新しいClipとして記録される場合、そのClip全体を参照するReal PlayListが新たに作成される操作である。

図4 Bは、Real PlayListのディバイド（divide：分割）に関する図であり、Real PlayListが所望な点で分けられて、2つのReal PlayListに分割される操作である。この分割という操作は、例えば、1つのPlayListにより管理される1つのクリップ内に、2つの番組が管理されているような場合に、ユーザが1つ1つの番組として登録（記録）し直したいといったようなときに行われる。この操作により、Clipの内容が変更される（Clip自体が分割される）ことはない。

図4 Cは、Real PlayListのコンバイン（combine：結合）に関する図であり、

2つのReal Playlistを結合して、1つの新しいReal Playlistにする操作である。この結合という操作は、例えば、ユーザが2つの番組を1つの番組として登録し直したいといったようなときに行われる。この操作により、Clipが変更される(Clip自体が1つにされる)ことはない。

図5 Aは、Real Playlist全体のデリート(delete:削除)に関する図であり、所定のReal Playlist全体を消去する操作がされた場合、削除されたReal Playlistが参照するClipの、対応するストリーム部分も削除される。

図5 Bは、Real Playlistの部分的な削除に関する図であり、Real Playlistの所望な部分が削除された場合、対応するPlayItemが、必要なClipのストリーム部分だけを参照するように変更される。そして、Clipの対応するストリーム部分は削除される。

図5 Cは、Real Playlistのミニマイズ(Minimize:最小化)に関する図であり、Real Playlistに対応するPlayItemを、Virtual Playlistに必要なClipのストリーム部分だけを参照するようにする操作である。Virtual Playlistにとって不必要的Clipの、対応するストリーム部分は削除される。

上述したような操作により、Real Playlistが変更されて、そのReal Playlistが参照するClipのストリーム部分が削除された場合、その削除されたClipを使用しているVirtual Playlistが存在し、そのVirtual Playlistにおいて、削除されたClipにより問題が生じる可能性がある。

そのようなことが生じないように、ユーザに、削除という操作に対して、「そのReal Playlistが参照しているClipのストリーム部分を参照しているVirtual Playlistが存在し、もし、そのReal Playlistが消去されると、そのVirtual Playlistもまた消去されることになるが、それでも良いか?」といったメッセージ等を表示させることにより、確認(警告)を促した後に、ユーザの指示により削除の処理を実行、又は、キャンセルする。又は、Virtual Playlistを削除する代わりに、Real Playlistに対してミニマイズの操作が行われるようにする。

次に、Virtual Playlistに対する操作について説明する。Virtual Playlistに対して操作が行われたとしても、Clipの内容が変更されることはない。図6 A及び図6 Bは、アセンブル(Assemble)編集(IN-OUT編集)に関する図で

あり、ユーザが見たいと所望した再生区間のPlayItemを作り、Virtual PlayListを作成するといった操作である。PlayItem間のシームレス接続が、アプリケーションフォーマットによりサポートされている（後述）。

図6 Aに示したように、2つのReal PlayList1, 2と、それぞれのReal PlayListに対応するClip1, 2が存在している場合に、ユーザがReal PlayList1内の所定の区間（In1乃至Out1までの区間：PlayItem1）を再生区間として指示し、続けて再生する区間として、Real PlayList2内の所定の区間（In2乃至Out2までの区間：PlayItem2）を再生区間として指示したとき、図6 Bに示すように、PlayItem1とPlayItem2から構成される1つのVirtual PlayListが作成される。

次に、Virtual PlayListの再編集（Re-editing）について説明する。再編集には、Virtual PlayListの中のイン点やアウト点の変更、Virtual PlayListへの新しいPlayItemの挿入（insert）や追加（append）、Virtual PlayListの中のPlayItemの削除等がある。また、Virtual PlayListそのものを削除することもできる。

図7は、Virtual PlayListへのオーディオのアフレコ（Audio dubbing（post recording））に関する図であり、Virtual PlayListへのオーディオのアフレコをサブバスとして登録する操作のことである。このオーディオのアフレコは、アプリケーションフォーマットによりサポートされている。Virtual PlayListのメインバスのAVストリームに、付加的なオーディオストリームが、サブバスとして付加される。

Real PlayListとVirtual PlayListで共通の操作として、図8に示すようなPlayListの再生順序の変更（Moving）がある。この操作は、ディスク（ボリューム）の中でのPlayListの再生順序の変更であり、アプリケーションフォーマットにおいて定義されるTable Of PlayList（図20等を参照して後述する）によってサポートされる。この操作により、Clipの内容が変更されるようなことはない。

次に、マーク（Mark）について説明する。マークは、Clip及びPlayListの中のハイライトや特徴的な時間を指定するために設けられている。Clipに付加されるマークは、AVストリームの内容に起因する特徴的なシーンを指定する、例えば、シーンチェンジ点等である。PlayListを再生するとき、そのPlayListが参照するClipのマークを参照して、使用することができる。

PlayListに付加されるマークは、主にユーザによってセットされる、例えば、ブックマークやリジューム点等である。Clip又はPlayListにマークをセットすることは、マークの時刻を示すタイムスタンプをマークリストに追加することにより行われる。また、マークを削除することは、マークリストの中から、そのマークのタイムスタンプを除去することである。したがって、マークの設定や削除により、AVストリームは何の変更もされない。

次にサムネイルについて説明する。サムネイルは、Volume、PlayList、及びClipに付加される静止画である。サムネイルには、2つの種類があり、1つは、内容を表す代表画としてのサムネイルである。これは主としてユーザがカーソル（不図示）等を操作して見たいものを選択するためのメニュー画面で使われるものである。もう1つは、マークが指しているシーンを表す画像である。

Volumeと各Playlistは代表画を持つことができるようとする必要がある。Volumeの代表画は、ディスク（記録媒体100、以下、記録媒体100はディスク状のものであるとし、適宜、ディスクと記述する）を記録再生装置1の所定の場所にセットしたときに、そのディスクの内容を表す静止画を最初に表示する場合等に用いられることを想定している。Playlistの代表画は、Playlistを選択するメニュー画面において、Playlistの内容を表すための静止画として用いられることを想定している。

Playlistの代表画として、Playlistの最初の画像をサムネイル（代表画）にすることが考えられるが、必ずしも再生時刻0の先頭の画像が内容を表す上で最適な画像とは限らない。そこで、Playlistのサムネイルとして、任意の画像をユーザが設定できるようにする。以上2種類のサムネイルをメニューサムネイルという。メニューサムネイルは頻繁に表示されるため、ディスクから高速に読み出される必要がある。このため、全てのメニューサムネイルを1つのファイルに格納することが効率的である。メニューサムネイルは、必ずしもボリューム内の動画から抜き出したピクチャである必要はなく、図10に示すように、パソコンコンピュータやデジタルスチルカメラから取り込まれた画像でもよい。

一方、ClipとPlaylistには、複数個のマークを打てる必要があり、マーク位置の内容を知るためにマーク点の画像を容易に見ることができるようにする必要が

ある。このようなマーク点を表すピクチャをマークサムネイル (Mark Thumbnail s) という。したがって、サムネイルの元となる画像は、外部から取り込んだ画像よりも、マーク点の画像を抜き出したものが主となる。

図 1 1 は、PlayListに付けられるマークと、そのマークサムネイルの関係について示す図であり、図 1 2 は、Clipに付けられるマークと、そのマークサムネイルの関係について示す図である。マークサムネイルは、メニューサムネイルと異なり、Playlistの詳細を表すときに、サブメニュー等で使われるため、短いアクセス時間で読み出されるようなことは要求されない。そのため、サムネイルが必要になる度に、記録再生装置 1 がファイルを開き、そのファイルの一部を読み出すことで多少時間がかかるても、問題にはならない。

また、ボリューム内に存在するファイル数を減らすために、全てのマークサムネイルは 1 つのファイルに格納するのがよい。Playlistはメニューサムネイル 1 つと複数のマークサムネイルを有することができるが、Clipは直接ユーザが選択する必要性がない（通常、Playlist経由で指定する）ため、メニューサムネイルを設ける必要はない。

図 1 3 は、上述したことを考慮した場合のメニューサムネイル、マークサムネイル、PlayList、及びClipの関係について示した図である。メニューサムネイルファイルには、PlayList毎に設けられたメニューサムネイルがファイルされている。メニューサムネイルファイルには、ディスクに記録されているデータの内容を代表するボリュームサムネイルが含まれている。マークサムネイルファイルは、各PlayList毎と各Clip毎に作成されたサムネイルがファイルされている。

次に、C P I (Characteristic Point Information) について説明する。C P I は、Clipインフォメーションファイルに含まれるデータであり、主に、それは Clipへのアクセスポイントのタイムスタンプが与えられたとき、Clip AV stream fileの中でデータの読み出しを開始すべきデータアドレスを見つけるために用いられる。本例では、2種類のC P I を用いる。1つは、EP_mapであり、もう一つは、TU_mapである。

EP_mapは、エントリポイント (EP) データのリストであり、それはエレメンタリーストリーム及びトランスポーテストリームから抽出されたものである。これ

は、AVストリームの中でデコードを開始すべきエントリポイントの場所を見つけるためのアドレス情報を持つ。1つのEPデータは、プレゼンテーションタイムスタンプ(PTS)と、そのPTSに対応するアクセスユニットのAVストリームの中のデータアドレスの対で構成される。

EP_mapは、主に2つの目的のために使用される。第1に、PlayListの中でプレゼンテーションタイムスタンプによって参照されるアクセスユニットのAVストリームの中のデータアドレスを見つけるために使用される。第2に、ファーストフォワード再生やファーストリバース再生のために使用される。記録再生装置1が、入力AVストリームを記録する場合、そのストリームのシンタクスを解析することができるとき、EP_mapが作成され、ディスクに記録される。

TU_mapは、デジタルインターフェースを通して入力されるトランスポートパケットの到着時刻に基づいたタイムユニット(TU)データのリストを持つ。これは、到着時刻ベースの時間とAVストリームの中のデータアドレスとの関係を与える。記録再生装置1が、入力AVストリームを記録する場合、そのストリームのシンタクスを解析することができないとき、TU_mapが作成され、ディスクに記録される。

STCInfoは、MPEG2トランスポートストリームをストアしているAVストリームファイルの中にあるSTCの不連続点情報をストアする。AVストリームがSTCの不連続点を持つ場合、そのAVストリームファイルの中で同じ値のPTSが現れるかもしれない。そのため、AVストリーム上のある時刻をPTSベースで指す場合、アクセスポイントのPTSだけではそのポイントを特定するためには不十分である。更に、そのPTSを含むところの連続なSTC区間のインデックスが必要である。連続なSTC区間を、このフォーマットではSTC-sequenceと呼び、そのインデックスをSTC-sequence-idと呼ぶ。STC-sequenceの情報は、Clip Information fileのSTCInfoで定義される。

STC-sequence-idは、EP_mapを持つAVストリームファイルで使用するものであり、TU_mapを持つAVストリームファイルではオプションである。

プログラムは、エレメンタリストリームの集まりであり、これらのストリームの同期再生のために、ただ1つのシステムタイムベースを共有するものである。

再生装置（図1の記録再生装置1）にとって、AVストリームのデコードに先だち、そのAVストリームの内容が分かることは有用である。例えば、ビデオやオーディオのエレメンタリーストリームを伝送するトランスポートパケットのPIDの値や、ビデオやオーディオのコンポーネント種類（例えば、HDTVのビデオとMPEG-2 AACのオーディオストリーム等）等の情報である。この情報はAVストリームを参照するところのPlayListの内容をユーザに説明するところのメニュー画面を作成するのに有用であるし、また、AVストリームのデコードに先だって、再生装置のAVデコーダ及びデマルチプレクサの初期状態をセットするために役立つ。

この理由のために、Clip Information fileは、プログラムの内容を説明するためのProgramInfoを持つ。

MPEG2トランスポートストリームをストアしているAVストリームファイルは、ファイルの中でプログラム内容が変化するかもしれない。例えば、ビデオエレメンタリーストリームを伝送するところのトランスポートパケットのPIDが変化したり、ビデオストリームのコンポーネント種類がSDTVからHDTVに変化する等である。ProgramInfoは、AVストリームファイルの中でのプログラム内容の変化点の情報をストアする。AVストリームファイルの中で、このフォーマットで定めるところのプログラム内容が一定である区間をProgram-sequenceと呼ぶ。

Program-sequenceは、EP_mapを持つAVストリームファイルで使用するものであり、TU_mapを持つAVストリームファイルではオプションである。

本例では、セルフエンコードのストリームフォーマット（SESF）を定義する。SESFは、アナログ入力信号を符号化する目的、及びディジタル入力信号（例えばDV）をデコードしてからMPEG2トランスポートストリームに符号化する場合に用いられる。

SESFは、MPEG-2トランスポートストリーム及びAVストリームについてのエレメンタリーストリームの符号化制限を定義する。記録再生装置1が、SESFストリームをエンコードし、記録する場合、EP_mapが作成され、ディスクに記録される。

デジタル放送のストリームは、次に示す方式の内のいずれかが用いられて記録媒体100に記録される。先ず、デジタル放送のストリームをSESFストリームにトランスコーディングする。この場合、記録されたストリームは、SESFに準拠しなければならない。この場合、EP_mapが作成されて、ディスクに記録されなければならない。

或いは、デジタル放送ストリームを構成するエレメンタリーストリームを新しいエレメンタリーストリームにトランスコーディングし、そのデジタル放送ストリームの規格化組織が定めるストリームフォーマットに準拠した新しいトランスポートストリームに再多重化する。この場合、EP_mapが作成されて、ディスクに記録されなければならない。

例えば、入力ストリームがISDB（日本のデジタルBS放送の規格名称）準拠のMPEG-2トランスポートストリームであり、それがHDTVビデオストリームとMPEG AACオーディオストリームを含むとする。HDTVビデオストリームをSDTVビデオストリームにトランスコーディングし、そのSDTVビデオストリームとオリジナルのAACオーディオストリームをTSに再多重化する。SDTVストリームと記録されるトランスポートストリームは、共にISDBフォーマットに準拠しなければならない。

デジタル放送のストリームが、記録媒体100に記録される際の他の方式として、入力トランスポートストリームをトランスペアレントに記録する（入力トランスポートストリームを何も変更しないで記録する）場合であり、そのときにEP_mapが作成されてディスクに記録される。

又は、入力トランスポートストリームをトランスペアレントに記録する（入力トランスポートストリームを何も変更しないで記録する）場合であり、そのときにTU_mapが作成されてディスクに記録される。

次にディレクトリとファイルについて説明する。以下、記録再生装置1をDVR (Digital Video Recording) と適宜記述する。図14はディスク上のディレクトリ構造の一例を示す図である。DVRのディスク上に必要なディレクトリは、図14に示したように、"DVR"ディレクトリを含むrootディレクトリ、"PLAYLIST"ディレクトリ、"CLIPINF"ディレクトリ、"M2TS"ディレクトリ、及び"DATA"ディ

レクトリを含む”DVR”ディレクトリである。rootディレクトリの下に、これら以外のディレクトリを作成されるようにしてもよいが、それらは、本例のアプリケーションフォーマットでは、無視されるとする。

”DVR”ディレクトリの下には、DVRアプリケーションフォーマットによって規定される全てのファイルとディレクトリがストアされる。”DVR”ディレクトリは、4個のディレクトリを含む。”PLAYLIST”ディレクトリの下には、Real PlayListとVirtual PlayListのデータベースファイルが置かれる。このディレクトリは、PlayListが1つもなくても存在する。

”CLIPINF”ディレクトリの下には、Clipのデータベースが置かれる。このディレクトリも、Clipが1つもなくても存在する。”M2TS”ディレクトリの下には、AVストリームファイルが置かれる。このディレクトリは、AVストリームファイルが1つもなくても存在する。”DATA”ディレクトリは、デジタルTV放送等のデータ放送のファイルがストアされる。

”DVR”ディレクトリは、次に示すファイルをストアする。”info.dvr”ファイルは、DVRディレクトリの下に作られ、アプリケーションレイヤの全体的な情報をストアする。DVRディレクトリの下には、ただ1つのinfo.dvrがなければならない。ファイル名は、info.dvrに固定されるとする。”menu.thmb”ファイルは、メニュー・サムネイル画像に関連する情報をストアする。DVRディレクトリの下には、0又は1つのメニュー・サムネイルがなければならない。ファイル名は、menu.thmbに固定されるとする。メニュー・サムネイル画像が1つもない場合、このファイルは、存在しなくてもよい。

”mark.thmb”ファイルは、マーク・サムネイル画像に関連する情報をストアする。DVRディレクトリの下には、0又は1つのマーク・サムネイルがなければならない。ファイル名は、mark.thmbに固定されるとする。メニュー・サムネイル画像が1つもない場合、このファイルは、存在しなくてもよい。

”PLAYLIST”ディレクトリは、2種類のPlaylistファイルをストアするものであり、それらは、Real PlaylistとVirtual Playlistである。”xxxxx.rpls”ファイルは、1つのReal Playlistに関連する情報をストアする。それぞれのReal Playlist毎に、1つのファイルが作られる。ファイル名は、”xxxxx.rpls”である。こ

ここで、"xxxxx"は、5個の0乃至9まで数字である。ファイル拡張子は、"rpls"でなければならぬとする。

"yyyyy.vpls"ファイルは、1つのVirtual PlayListに関連する情報をストアする。それぞれのVirtual PlayList毎に、1つのファイルが作られる。ファイル名は、"yyyyy.vpls"である。ここで、"yyyyy"は、5個の0乃至9まで数字である。ファイル拡張子は、"vpls"でなければならぬとする。

"CLIPINF"ディレクトリは、それぞれのAVストリームファイルに対応して、1つのファイルをストアする。"zzzzz.clpi"ファイルは、1つのAVストリームファイル(Clip AV stream file又はBridge-Clip AV stream file)に対応するClip Information fileである。ファイル名は、"zzzzz.clpi"であり、"zzzzz"は、5個の0乃至9までの数字である。ファイル拡張子は、"clpi"でなければならぬとする。

"M2TS"ディレクトリは、AVストリームのファイルをストアする。"zzzzz.m2ts"ファイルは、DVRシステムにより扱われるAVストリームファイルである。これは、Clip AV stream file又はBridge-Clip AV streamである。ファイル名は、"zzzzz.m2ts"であり、"zzzzz"は、5個の0乃至9までの数字である。ファイル拡張子は、"m2ts"でなければならぬとする。

"DATA"ディレクトリは、データ放送から伝送されるデータをストアするものであり、データとは、例えば、XML fileやMHEGファイル等である。

次に、各ディレクトリ(ファイル)のシンタクスとセマンティクスを説明する。先ず、"info.dvr"ファイルについて説明する。図15は、"info.dvr"ファイルのシンタクスを示す図である。"info.dvr"ファイルは、3個のオブジェクトから構成され、それらは、DVRVolume()、TableOfPlayLists()、及びMakerPrivateData()である。

図15に示したinfo.dvrのシンタクスについて説明すると、TableOfPlayLists_Start_addressは、info.dvrファイルの先頭のバイトからの相対バイト数を単位として、TableOfPlayList()の先頭アドレスを示す。相対バイト数は0からカウントされる。

MakerPrivateData_Start_addressは、info.dvrファイルの先頭のバイトからの

相対バイト数を単位として、`MakerPrivateData()` の先頭アドレスを示す。相対バイト数は 0 からカウントされる。`padding_word` (パディングワード) は、`info.dvr` のシンタクスに従って挿入される。N 1 と N 2 は、0 又は任意の正の整数である。それぞれのパディングワードは、任意の値を取るようにしてよい。

`DVRVolume()` は、ボリューム (ディスク) の内容を記述する情報をストアする。図 1.6 は、`DVRVolume()` のシンタクスを示す図である。図 1.6 に示した `DVRVolume()` のシンタクスを説明すると、`version_number` は、この `DVRVolume()` のバージョンナンバを示す 4 個のキャラクター文字を示す。`version_number` は、IS 0 646 に従って、"0045" と符号化される。

`length` は、この `length` フィールドの直後から `DVRVolume()` の最後までの `DVRVolume()` のバイト数を示す 32 ビットの符号なし整数で表される。

`ResumeVolume()` は、ボリュームの中で最後に再生した Real PlayList 又は Virtual PlayList のファイル名を記憶している。但し、Real PlayList 又は Virtual PlayList の再生をユーザが中断したときの再生位置は、`PlaylistMark()` において定義される `resume-mark` にストアされる。

図 1.7 は、`ResumeVolume()` のシンタクスを示す図である。図 1.7 に示した `ResumeVolume()` のシンタクスを説明すると、`valid_flag` は、この 1 ビットのフラグが 1 にセットされている場合、`resume_PlayList_name` フィールドが有効であることを示し、このフラグが 0 にセットされている場合、`resume_PlayList_name` フィールドが無効であることを示す。

`resume_PlayList_name` の 10 バイトのフィールドは、リジュームされるべき Real PlayList 又は Virtual PlayList のファイル名を示す。

図 1.6 に示した `DVRVolume()` のシンタクスのなかの、`UIAppInfoVolume` は、ボリュームについてのユーザインターフェースアプリケーションのパラメータをストアする。図 1.8 は、`UIAppInfoVolume` のシンタクスを示す図であり、そのセマンティクスを説明すると、`character_set` の 8 ビットのフィールドは、`Volume_name` フィールドに符号化されているキャラクター文字の符号化方法を示す。その符号化方法は、図 1.9 に示される値に対応する。

`name_length` の 8 ビットフィールドは、`Volume_name` フィールドの中に示される

ボリューム名のバイト長を示す。Volume_nameのフィールドは、ボリュームの名称を示す。このフィールドの中の左からname_length数のバイト数が、有効なキャラクター文字であり、それはボリュームの名称を示す。Volume_nameフィールドの中で、それら有効なキャラクター文字の後の値は、どんな値が入っていてもよい。

Volume_protect_flagは、ボリュームの中のコンテンツを、ユーザに制限することなしに見せてよいかどうかを示すフラグである。このフラグが1にセットされている場合、ユーザが正しくPIN番号（パスワード）を入力できたときだけ、そのボリュームのコンテンツを、ユーザに見せること（再生されること）が許可される。このフラグが0にセットされている場合、ユーザがPIN番号を入力しなくても、そのボリュームのコンテンツを、ユーザに見せることが許可される。

最初に、ユーザが、ディスクをプレーヤへ挿入した時点において、もしこのフラグが0にセットされているか、又は、このフラグが1にセットされてもユーザがPIN番号を正しく入力できたならば、記録再生装置1は、そのディスクの中のPlayListの一覧を表示させる。それぞれのPlayListの再生制限は、volume_protect_flagとは無関係であり、それはUIAppInfoPlayList()の中に定義されるplayback_control_flagによって示される。

PINは、4個の0乃至9までの数字で構成され、それぞれの数字は、ISO／IEC 646に従って符号化される。ref_thumbnail_indexのフィールドは、ボリュームに付加されるサムネイル画像の情報を示す。ref_thumbnail_indexフィールドが、0xFFFFでない値の場合、そのボリュームにはサムネイル画像が付加されており、そのサムネイル画像は、menu.thumファイルの中にストアされている。その画像は、menu.thumファイルの中でref_thumbnail_indexの値を用いて参照される。ref_thumbnail_indexフィールドが、0xFFFFである場合、そのボリュームにはサムネイル画像が付加されていないことを示す。

rp_info_valid_flagは、これが1である場合に次に続くrp_ref_to_PlayList_file_name, rp_ref_to_PlayItem_id及びrp_time_stampが有効な値を持つこと示す。

rp_ref_to_PlayList_file_nameは、上記のボリュームを代表するメニュー・サムネイルが、あるPlayList中の画像から抜き出した画像から作られていることを示し、そのPlayListファイルの名前を示す。

`rp_ref_to_PlayItem_id`は、`rp_ref_to_PlayList_file_name`で示されるPlayListの中の1つのPlayItemを指すPlayItem_idを示し、且つ、上記のボリュームを代表するメニューサムネイルが、そのPlayItem中の画像から抜き出した画像から作られていることを示す。

`rp_time_stamp`は、`rp_ref_to_PlayItem_id`が指すPlayItem中の1つの画像のプレゼンテーションタイムスタンプを示し、且つ、その画像から上記のボリュームを代表するメニューサムネイルが作られていることを示す。

次に図1-5に示したinfo.dvrのシンタクス内のTableOfPlayLists()について説明する。TableOfPlayLists()は、PlayList(Real PlayListとVirtual PlayList)のファイル名をストアする。ボリュームに記録されている全てのPlayListファイルは、TableOfPlayList()の中に含まれる。TableOfPlayLists()は、ボリュームの中のPlayListのデフォルトの再生順序を示す。

図20は、TableOfPlayLists()のシンタクスを示す図であり、そのシンタクスについて説明すると、TableOfPlayListsのversion_numberは、このTableOfPlayListsのバージョンナンバを示す4個のキャラクター文字を示す。version_numberは、ISO 646に従って、"0045"と符号化されなければならない。

`length`は、この`length`フィールドの直後から`TableOfPlayLists()`の最後までの`TableOfPlayLists()`のバイト数を示す32ビットの符号なしの整数である。`number_of_PlayLists`の16ビットのフィールドは、`Playlist_file_name`を含む`for-loop`のループ回数を示す。この数字は、ボリュームに記録されている`Playlist`の数に等しくなければならない。`Playlist_file_name`の10バイトの数字は、`Playlist`のファイル名を示す。

図21は、TableOfPlayLists()のシンタクスを別実施の構成を示す図である。図21に示したシンタクスは、図20に示したシンタクスに、UIAppinfoPlayList（後述）を含ませた構成とされている。このように、UIAppinfoPlayListを含ませた構成とすることで、TableOfPlayListsを読み出すだけで、メニュー画面を作成することが可能となる。ここでは、図20に示したシンタクスを用いるとして以下の説明をする。

図1.5に示したinfo.dvrのシンタクス内のMakersPrivateDataについて説明する。

MakersPrivateDataは、記録再生装置1のメーカーが、各社の特別なアプリケーションのために、MakersPrivateData () の中にメーカーのプライベートデータを挿入できるように設けられている。各メーカーのプライベートデータは、それを定義したメーカーを識別するために標準化されたmaker_IDを持つ。MakersPrivateData () は、1つ以上のmaker_IDを含んでもよい。

所定のメーカーが、プライベートデータを挿入したいときに、既に他のメーカーのプライベートデータがMakersPrivateData () に含まれていた場合、他のメーカーは、既にある古いプライベートデータを消去するのではなく、新しいプライベートデータをMakersPrivateData () の中に追加するようにする。このように、本例においては、複数のメーカーのプライベートデータが、1つのMakersPrivateData () に含まれることが可能であるようにする。

図22は、MakersPrivateDataのシンタクスを示す図である。図22に示したMakersPrivateDataのシンタクスについて説明すると、version_numberは、このMakersPrivateData () のバージョンナンバを示す4個のキャラクター文字を示す。version_numberは、ISO 646に従って、"0045"と符号化されなければならない。lengthは、このlengthフィールドの直後からMakersPrivateData () の最後までのMakersPrivateData () のバイト数を示す32ビットの符号なし整数を示す。

mpd_blocks_start_addressは、MakersPrivateData () の先頭のバイトからの相対バイト数を単位として、最初のmpd_block () の先頭バイトアドレスを示す。相対バイト数は0からカウントされる。number_of_maker_entriesは、MakersPrivateData () の中に含まれているメーカープライベートデータのエントリ数を与える16ビットの符号なし整数である。MakersPrivateData () の中に、同じmaker_IDの値を持つメーカープライベートデータが2個以上存在してはならない。

mpd_block_sizeは、1024バイトを単位として、1つのmpd_blockの大きさを与える16ビットの符号なし整数である。例えば、mpd_block_size=1ならば、それは1つのmpd_blockの大きさが1024バイトであることを示す。number_of_mpd_blocksは、MakersPrivateData () の中に含まれるmpd_blockの数を与える16ビットの符号なし整数である。maker_IDは、そのメーカープライベートデータを作成したDVRシステムの製造メーカーを示す16ビットの符号なし整数である。ma

ker_IDに符号化される値は、このDVRフォーマットのライセンサによって指定される。

maker_model_codeは、そのメーカプライベートデータを作成したDVRシステムのモデルナンバーコードを示す16ビットの符号なし整数である。maker_model_codeに符号化される値は、このフォーマットのライセンスを受けた製造メーカーによって設定される。start_mpd_block_numberは、そのメーカプライベートデータが開始されるmpd_blockの番号を示す16ビットの符号なし整数である。メーカプライベートデータの先頭データは、mpd_blockの先頭にアラインされなければならない。start_mpd_block_numberは、mpd_blockのfor-loopの中の変数jに対応する。

mpd_lengthは、バイト単位でメーカプライベートデータの大きさを示す32ビットの符号なし整数である。mpd_blockは、メーカプライベートデータがストアされる領域である。MakersPrivateData()の中の全てのmpd_blockは、同じサイズでなければならない。

次に、Real PlayList fileとVirtual PlayList fileについて、換言すれば、xxxx.rplsとyyyyy.vplsについて説明する。図23は、xxxx.rpls(Real PlayList)、又は、yyyyy.vpls(Virtual PlayList)のシンタクスを示す図である。xxx.rplsとyyyyy.vplsは、同一のシンタクス構成を持つ。xxxx.rplsとyyyyy.vplsは、それぞれ、3個のオブジェクトから構成され、それらは、Playlist()、PlaylistMark()、及びMakerPrivateData()である。

PlaylistMark_Start_addressは、Playlistファイルの先頭のバイトからの相対バイト数を単位として、PlaylistMark()の先頭アドレスを示す。相対バイト数は0からカウントされる。

MakerPrivateData_Start_addressは、Playlistファイルの先頭のバイトからの相対バイト数を単位として、MakerPrivateData()の先頭アドレスを示す。相対バイト数は0からカウントされる。

padding_word(パディングワード)は、Playlistファイルのシンタクスに従つて挿入され、N1とN2は、0又は任意の正の整数である。それぞれのパディングワードは、任意の値を取るようにしてよい。

ここで、既に、簡便に説明したが、Playlistについて更に説明する。ディスク

内にある全てのReal Playlistによって、Bridge-Clip（後述）を除く全てのClipの中の再生区間が参照されていなければならない。且つ、2つ以上のReal Playlistが、それらのPlayItemで示される再生区間を同一のClipの中でオーバーラップさせてはならない。

図24A～図24Cを参照して更に説明すると、図24Aに示したように、全てのClipは、対応するReal Playlistが存在する。この規則は、図24Bに示したように、編集作業が行われた後においても守られる。したがって、全てのClipは、どれかしらのReal Playlistを参照することにより、必ず視聴することが可能である。

図24Cに示したように、Virtual Playlistの再生区間は、Real Playlistの再生区間又はBridge-Clipの再生区間の中に含まれていなければならない。どのVirtual Playlistにも参照されないBridge-Clipがディスクの中に存在してはならない。

Real Playlistは、PlayItemのリストを含むが、SubPlayItemを含んではならない。Virtual Playlistは、PlayItemのリストを含み、Playlist()の中に示されるCPI_typeがEP_map typeであり、且つPlaylist_typeが0（ビデオとオーディオを含むPlaylist）である場合、Virtual Playlistは、1つのSubPlayItemを含むことができる。本例におけるPlaylist()では、SubPlayItemはオーディオのアフレコの目的にだけに使用される、そして、1つのVirtual Playlistが持つSubPlayItemの数は、0又は1でなければならない。

次に、Playlistについて説明する。図25は、Playlistのシンタクスを示す図である。図25に示したPlaylistのシンタクスを説明すると、version_numberは、このPlaylist()のバージョンナンバを示す4個のキャラクター文字である。version_numberは、ISO 646に従って、"0045"と符号化されなければならない。lengthは、このlengthフィールドの直後からPlaylist()の最後までのPlaylist()のバイト数を示す32ビットの符号なし整数である。Playlist_typeは、このPlaylistのタイプを示す8ビットのフィールドであり、その一例を図26に示す。

CPI_typeは、1ビットのフラグであり、PlayItem()及びSubPlayItem()によって参照されるClipのCPI_typeの値を示す。1つのPlaylistによって参照される

全てのClipは、それらのCPI () の中に定義されるCPI_typeの値が同じでなければならぬ。number_of_PlayItemsは、PlayListの中にあるPlayItemの数を示す16ビットのフィールドである。

所定のPlayItem () に対応するPlayItem_idは、PlayItem () を含むfor-loopの中で、そのPlayItem () の現れる順番により定義される。PlayItem_idは、0から開始される。number_of_SubPlayItemsは、PlayListの中にあるSubPlayItemの数を示す16ビットのフィールドである。この値は、0又は1である。付加的なオーディオストリームのバス（オーディオストリームバス）は、サブバスの一種である。

次に、図25に示したPlayListのシンタクスのUIAppInfoPlaylistについて説明する。UIAppInfoPlaylistは、PlayListについてのユーザインタフェースアプリケーションのパラメータをストアする。図27は、UIAppInfoPlaylistのシンタクスを示す図である。図27に示したUIAppInfoPlaylistのシンタクスを説明すると、character_setは、8ビットのフィールドであり、Playlist_nameフィールドに符号化されているキャラクター文字の符号化方法を示す。その符号化方法は、図19に示したテーブルに準拠する値に対応する。

name_lengthは、8ビットフィールドであり、Playlist_nameフィールドの中に示されるPlayList名のバイト長を示す。Playlist_nameのフィールドは、PlayListの名称を示す。このフィールドの中の左からname_length数のバイト数が、有効なキャラクター文字であり、それはPlayListの名称を示す。Playlist_nameフィールドの中で、それら有効なキャラクター文字の後の値は、どんな値が入っていてもよい。

record_time_and_dateは、PlayListが記録されたときの日時をストアする56ビットのフィールドである。このフィールドは、年／月／日／時／分／秒について、14個の数字を4ビットのBinary Coded Decimal (BCD) で符号化したものである。例えば、2001/12/23:01:02:03は、“0x20011223010203”と符号化される。

durationは、PlayListの総再生時間を時間／分／秒の単位で示した24ビットのフィールドである。このフィールドは、6個の数字を4ビットのBinary Coded

Decimal (B C D) で符号化したものである。例えば、01:45:30は、“0x014530”と符号化される。

valid_periodは、PlayListが有効である期間を示す32ビットのフィールドである。このフィールドは、8個の数字を4ビットのBinary Coded Decimal (B C D) で符号化したものである。例えば、記録再生装置1は、この有効期間の過ぎたPlayListを自動消去する、といったように用いられる。例えば、2001/05/07 は、“0x20010507”と符号化される。

maker_idは、そのPlayListを最後に更新したDVRプレーヤ（記録再生装置1）の製造者を示す16ビットの符号なし整数である。maker_idに符号化される値は、DVRフォーマットのライセンサによって割り当てられる。maker_codeは、そのPlayListを最後に更新したDVRプレーヤのモデル番号を示す16ビットの符号なし整数である。maker_codeに符号化される値は、DVRフォーマットのライセンスを受けた製造者によって決められる。

playback_control_flagのフラグが1にセットされている場合、ユーザが正しくPIN番号を入力できた場合にだけ、そのPlayListは再生される。このフラグが0にセットされている場合、ユーザがPIN番号を入力しなくても、ユーザは、そのPlayListを視聴することができる。

write_protect_flagは、図28Aにテーブルを示すように、1にセットされている場合、write_protect_flagを除いて、そのPlayListの内容は、消去及び変更されない。このフラグが0にセットされている場合、ユーザは、そのPlayListを自由に消去及び変更できる。このフラグが1にセットされている場合、ユーザが、そのPlayListを消去、編集、又は上書きする前に、記録再生装置1はユーザに再確認するようなメッセージを表示させる。

write_protect_flagが0にセットされているReal PlayListが存在し、且つ、そのReal PlayListのClipを参照するVirtual PlayListが存在し、そのVirtual PlayListのwrite_protect_flagが1にセットされていてもよい。ユーザが、Real PlayListを消去しようとする場合、記録再生装置1は、そのReal PlayListを消去する前に、上記Virtual PlayListの存在をユーザに警告するか、又は、そのReal PlayListを“Minimize”する。

`is_played_flag`は、図28Bに示すように、フラグが1にセットされている場合、その`PlayList`は、記録されてから1度は再生されたことを示し、0にセットされている場合、その`PlayList`は、記録されてから1度も再生されたことがないことを示す。

`archive`は、図28Cに示すように、その`PlayList`がオリジナルであるか、コピーされたものであるかを示す2ビットのフィールドである。`ref_thumbnail_index`のフィールドは、`PlayList`を代表するサムネイル画像の情報を示す。`ref_thumbnail_index`フィールドが、0xFFFFでない値の場合、その`PlayList`には、`PlayList`を代表するサムネイル画像が付加されており、そのサムネイル画像は、`menu.thum`ファイルの中にストアされている。その画像は、`menu.thum`ファイルの中で`ref_thumbnail_index`の値を用いて参照される。`ref_thumbnail_index`フィールドが、0xFFFFである場合、その`PlayList`には、`PlayList`を代表するサムネイル画像が付加されていない。

次に`PlayItem`について説明する。1つの`PlayItem()`は、基本的に次のデータを含む。`Clip`のファイル名を指定するための`Clip_information_file_name`、`Clip`の再生区間を特定するための`IN_time`と`OUT_time`のペア、`PlayList()`において定義される`CPI_type`が`EP_map type`である場合、`IN_time`と`OUT_time`が参照するところの`STC_sequence_id`、及び、先行する`PlayItem`と現在の`PlayItem`との接続の状態を示すところの`connection_condition`である。

`PlayList`が2つ以上の`PlayItem`から構成されるとき、それらの`PlayItem`は`PlayList`のグローバル時間軸上に、時間のギャップ又はオーバーラップなしに1列に並べられる。`PlayList()`において定義される`CPI_type`が`EP_map type`であり、且つ現在の`PlayItem`が`BridgeSequence()`を持たないとき、その`PlayItem`において定義される`IN_time`と`OUT_time`のペアは、`STC_sequence_id`によって指定される同じS T C連続区間上の時間を指していなければならない。そのような例を図29に示す。

図30は、`PlayList()`において定義される`CPI_type`が`EP_map type`であり、且つ現在の`PlayItem`が`BridgeSequence()`を持つとき、次に説明する規則が適用される場合を示している。現在の`PlayItem`に先行する`PlayItem`の`IN_time`（図の中

でIN_time1と示されているもの)は、先行するPlayItemのSTC_sequence_idによって指定されるS T C連続区間上の時間を指している。先行するPlayItemのOUT_time(図の中でOUT_time1と示されているもの)は、現在のPlayItemのBridgeSequenceInfo()の中で指定されるBridge-Clipの中の時間を指している。このOUT_timeは、後述する符号化制限に従っていなければならない。

現在のPlayItemのIN_time(図の中でIN_time2と示されているもの)は、現在のPlayItemのBridgeSequenceInfo()の中で指定されるBridge-Clipの中の時間を指している。このIN_timeも、後述する符号化制限に従っていなければならない。現在のPlayItemのPlayItemのOUT_time(図の中でOUT_time2と示されているもの)は、現在のPlayItemのSTC_sequence_idによって指定されるS T C連続区間上の時間を指している。

図31に示すように、Playlist()のCPI_typeがTU_map typeである場合、PlayItemのIN_timeとOUT_timeのペアは、同じClip AVストリーム上の時間を指している。

PlayItemのシンタクスは、図32に示すようになる。図32に示したPlayItemのシンタクスを説明すると、Clip_Information_file_nameのフィールドは、Clip Information fileのファイル名を示す。このClip Information fileのClipInfo()において定義されるClip_stream_typeは、Clip AV streamを示していなければならない。

STC_sequence_idは、8ビットのフィールドであり、PlayItemが参照するS T C連続区間のSTC_sequence_idを示す。Playlist()の中で指定されるCPI_typeがTU_map typeである場合、この8ビットフィールドは何も意味を持たず、0にセットされる。IN_timeは、32ビットフィールドであり、PlayItemの再生開始時刻をストアする。IN_timeのセマンティクスは、図33に示すように、Playlist()において定義されるCPI_typeによって異なる。

OUT_timeは、32ビットフィールドであり、PlayItemの再生終了時刻をストアする。OUT_timeのセマンティクスは、図34に示すように、Playlist()において定義されるCPI_typeによって異なる。

Connection_Conditionは、図35に示したような先行するPlayItemと、現在の

PlayItemとの間の接続状態を示す2ビットのフィールドである。図36A～図36Dは、図35に示したConnection_Conditionの各状態について説明する図である。

次に、BridgeSequenceInfoについて、図37を参照して説明する。BridgeSequenceInfo()は、現在のPlayItemの付属情報であり、次に示す情報を持つ。Bridge-Clip AV streamファイルとそれに対応するClip Information fileを指定するBridge_Clip_Information_file_nameを含む。

また、先行するPlayItemが参照するClip AV stream上のソースパケットのアドレスであり、このソースパケットに続いてBridge-Clip AV streamファイルの最初のソースパケットが接続される。このアドレスは、RSPN_exit_from_previous_Clipと称される。更に現在のPlayItemが参照するClip AV stream上のソースパケットのアドレスであり、このソースパケットの前にBridge-Clip AV streamファイルの最後のソースパケットが接続される。このアドレスは、RSPN_enter_to_current_Clipと称される。

図37において、RSPN_arrival_time_discontinuityは、the Bridge-Clip AV streamファイルの中でアライバルタイムベースの不連続点があるところのソースパケットのアドレスを示す。このアドレスは、ClipInfo()の中において定義される。

図38は、BridgeSequenceinfoのシンタクスを示す図である。図38に示したBridgeSequenceinfoのシンタクスを説明すると、Bridge_Clip_Information_file_nameのフィールドは、Bridge-Clip AV streamファイルに対応するClip Information fileのファイル名を示す。このClip Information fileのClipInfo()において定義されるClip_stream_typeは、'Bridge-Clip AV stream'を示していなければならない。

RSPN_exit_from_previous_Clipの32ビットフィールドは、先行するPlayItemが参照するClip AV stream上のソースパケットの相対アドレスであり、このソースパケットに続いてBridge-Clip AV streamファイルの最初のソースパケットが接続される。RSPN_exit_from_previous_Clipは、ソースパケット番号を単位とする大きさであり、先行するPlayItemが参照するClip AV streamファイルの最初のソ

ースパケットからClipInfo () において定義されるoffset_SPNの値を初期値としてカウントされる。

RSPN_enter_to_current_Clipの32ビットフィールドは、現在のPlayItemが参照するClip AV stream上のソースパケットの相対アドレスであり、このソースパケットの前にBridge-Clip AV streamファイルの最後のソースパケットが接続される。RSPN_exit_from_previous_Clipは、ソースパケット番号を単位とする大きさであり、現在のPlayItemが参照するClip AV streamファイルの最初のソースパケットからClipInfo () において定義されるoffset_SPNの値を初期値としてカウントされる。

次に、SubPlayItemについて、図39を参照して説明する。SubPlayItem () の使用は、PlayList () のCPI_typeがEP_map typeである場合だけに許される。本例においては、SubPlayItemはオーディオのアフレコの目的のためだけに使用されるとする。SubPlayItem () は、次に示すデータを含む。先ず、PlayListの中のsub pathが参照するClipを指定するためのClip_information_file_nameを含む。

また、Clipの中のsub pathの再生区間を指定するためのSubPath_IN_timeとSubPath_OUT_timeを含む。更に、main pathの時間軸上でsub pathが再生開始する時刻を指定するためのsync_PlayItem_idとsync_start PTS_of_PlayItemを含む。sub pathに参照されるオーディオのClip AV streamは、STC不連続点（システムタイムベースの不連続点）を含んではならない。sub pathに使われるClipのオーディオサンプルのクロックは、main pathのオーディオサンプルのクロックにロックされている。

図40は、SubPlayItemのシンタクスを示す図である。図40に示したSubPlayItemのシンタクスを説明すると、Clip_Information_file_nameのフィールドは、Clip Information fileのファイル名を示し、それはPlayListの中でsub pathによって使用される。このClip Information fileのClipInfo () において定義されるClip_stream_typeは、Clip AV streamを示していなければならない。

SubPath_typeの8ビットのフィールドは、sub pathのタイプを示す。ここでは、図41に示すように、'0x00'しか設定されておらず、他の値は、将来のために確保されている。

`sync_PlayItem_id`の8ビットのフィールドは、`main path`の時間軸上で`sub path`が再生開始する時刻が含まれる`PlayItem`の`PlayItem_id`を示す。所定の`PlayItem`に対応する`PlayItem_id`の値は、`PlayList()`において定義される（図25参照）。

`sync_start PTS_of_PlayItem`の32ビットのフィールドは、`main path`の時間軸上で`sub path`が再生開始する時刻を示し、`sync_PlayItem_id`で参照される`PlayItem`上のPTS（Presentation Time Stamp）の上位32ビットを示す。`SubPath_IN_time`の32ビットフィールドは、`Sub path`の再生開始時刻をストアする。`SubPath_OUT_time`は、`Sub Path`の中で最初のプレゼンテーションユニットに対応する33ビット長のPTSの上位32ビットを示す。

`SubPath_OUT_time`の32ビットフィールドは、`Sub path`の再生終了時刻をストアする。`SubPath_OUT_time`は、次式によって算出される`Presentation_end_TS`の値の上位32ビットを示す。

$$\text{Presentation_end_TS} = \text{PTS_out} + \text{AU_duration}$$

ここで、`PTS_out`は、`SubPath`の最後のプレゼンテーションユニットに対応する33ビット長のPTSである。`AU_duration`は、`SubPath`の最後のプレゼンテーションユニットの90kHz単位の表示期間である。

次に、図23に示したxxxx.rplsとyyyy.vplsのシンタクス内の`PlayListMark()`について説明する。`PlayList`についてのマーク情報は、この`PlayListMark`にストアされる。図42は、`PlayListMark`のシンタクスを示す図である。図42に示した`PlayListMark`のシンタクスについて説明すると、`version_number`は、この`PlayListMark()`のバージョンナンバを示す4個のキャラクター文字である。`version_number`は、ISO 646に従って、"0045"と符号化されなければならない。

`length`は、この`length`フィールドの直後から`PlayListMark()`の最後までの`PlayListMark()`のバイト数を示す32ビットの符号なし整数である。`number_of_PlayList_marks`は、`PlayListMark`の中にストアされているマークの個数を示す1~6ビットの符号なし整数である。`number_of_PlayList_marks`は、0であってよい。`mark_type`は、マークのタイプを示す8ビットのフィールドであり、図43に示すテーブルに従って符号化される。

`mark_time_stamp`の32ビットフィールドは、マークが指定されたポイントを示す

タイムスタンプをストアする。`mark_time_stamp`のセマンティクスは、図44に示すように、`PlayList()`において定義されるCPI_typeによって異なる。`PlayItem_id`は、マークが置かれているところの`PlayItem`を指定する8ビットのフィールドである。所定の`PlayItem`に対応する`PlayItem_id`の値は、`PlayList()`において定義される（図25参照）。

`character_set`の8ビットのフィールドは、`mark_name`フィールドに符号化されているキャラクター文字の符号化方法を示す。その符号化方法は、図19に示した値に対応する。`name_length`の8ビットフィールドは、`Mark_name`フィールドの中に示されるマーク名のバイト長を示す。`mark_name`のフィールドは、マークの名称を示す。このフィールドの中の左から`name_length`数のバイト数が、有効なキャラクター文字であり、それはマークの名称を示す。`Mark_name`フィールドの中で、それら有効なキャラクター文字の後の値は、どのような値が設定されてもよい。

`ref_thumbnail_index`のフィールドは、マークに付加されるサムネイル画像の情報を示す。`ref_thumbnail_index`フィールドが、0xFFFFでない値の場合、そのマークにはサムネイル画像が付加されており、そのサムネイル画像は、`mark.thmb`ファイルの中にストアされている。その画像は、`mark.thmb`ファイルの中で`ref_thumbnail_index`の値を用いて参照される（後述）。`ref_thumbnail_index`フィールドが、0xFFFFである場合、そのマークにはサムネイル画像が付加されていないことを示す。

次に、`Clip information file`について説明する。`zzzzz.clpi` (`Clip information file`ファイル) は、図45に示すように6個のオブジェクトから構成される。それらは、`ClipInfo()`、`STC_Info()`、`ProgramInfo()`、`CPI()`、`ClipMark()`、及び`MakerPrivateData()`である。AVストリーム (`Clip AV Stream`又は`Bridge-Clip AV stream`) とそれに対応する`Clip Information`ファイルは、同じ数字列の”`zzzzz`”が使用される。

図45に示した`zzzzz.clpi` (`Clip information file`ファイル) のシンタクスについて説明すると、`ClipInfo_Start_address`は、`zzzzz.clpi`ファイルの先頭のバイトからの相対バイト数を単位として、`ClipInfo()`の先頭アドレスを示す。相対バイト数は0からカウントされる。

STC_Info_Start_addressは、zzzzz.clpiファイルの先頭のバイトからの相対バイト数を単位として、STC_Info () の先頭アドレスを示す。相対バイト数は0からカウントされる。ProgramInfo_Start_addressは、zzzzz.clpiファイルの先頭のバイトからの相対バイト数を単位として、ProgramInfo () の先頭アドレスを示す。相対バイト数は0からカウントされる。CPI_Start_addressは、zzzzz.clpiファイルの先頭のバイトからの相対バイト数を単位として、CPI () の先頭アドレスを示す。相対バイト数は0からカウントされる。

ClipMark_Start_addressは、zzzzz.clpiファイルの先頭のバイトからの相対バイト数を単位として、ClipMark () の先頭アドレスを示す。相対バイト数は0からカウントされる。MakerPrivateData_Start_addressは、zzzzz.clpiファイルの先頭のバイトからの相対バイト数を単位として、MakerPrivateData () の先頭アドレスを示す。相対バイト数は0からカウントされる。padding_word（パディングワード）は、zzzzz.clpiファイルのシンタクスに従って挿入される。N1, N2, N3, N4、及びN5は、0又は任意の正の整数でなければならない。それぞれのパディングワードは、任意の値がとられるようにしてもよい。

次に、ClipInfoについて説明する。図4 6は、ClipInfoのシンタクスを示す図である。ClipInfo () は、それに対応するAVストリームファイル（Clip AVストリーム又はBridge-Clip AVストリームファイル）の属性情報をストアする。

図4 6に示したClipInfoのシンタクスについて説明すると、version_numberは、このClipInfo () のバージョンナンバを示す4個のキャラクター文字である。version_numberは、ISO 646に従って、"0045"と符号化されなければならない。lengthは、このlengthフィールドの直後からClipInfo () の最後までのClipInfo () のバイト数を示す32ビットの符号なし整数である。Clip_stream_typeの8ビットのフィールドは、図4 7に示すように、Clip Informationファイルに対応するAVストリームのタイプを示す。それぞれのタイプのAVストリームのストリームタイプについては後述する。

offset_SPNの32ビットのフィールドは、AVストリーム（Clip AVストリーム又はBridge-Clip AVストリーム）ファイルの最初のソースパケットについてのソースパケット番号のオフセット値を与える。AVストリームファイルが最初

にディスクに記録されるとき、このoffset_SPNは0でなければならない。

図48に示すように、AVストリームファイルのはじめの部分が編集によって消去されたとき、offset_SPNは、0以外の値をとってもよい。本例では、offset_SPNを参照する相対ソースパケット番号（相対アドレス）が、しばしば、RSPN_x_xx（xxxは変形する。例。RSPN_EP_start）の形式でシンタクスの中に記述されている。相対ソースパケット番号は、ソースパケット番号を単位とする大きさであり、AVストリームファイルの最初のソースパケットからoffset_SPNの値を初期値としてカウントされる。

AVストリームファイルの最初のソースパケットから相対ソースパケット番号で参照されるソースパケットまでのソースパケットの数（SPN_xxx）は、次式で算出される。

$$\text{SPN}_\text{xxx} = \text{RSPN}_\text{xxx} - \text{offset}_\text{SPN}$$

図48に、offset_SPNが4である場合の例を示す。

TS_recording_rateは、24ビットの符号なし整数であり、この値は、DVRドライブ（書込部22）へ又はDVRドライブ（読み出部28）からのAVストリームの必要な入出力のビットレートを与える。record_time_and_dateは、Clipに対応するAVストリームが記録されたときの日時をストアする56ビットのフィールドであり、年／月／日／時／分／秒について、14個の数字を4ビットのBinary Coded Decimal (BCD)で符号化したものである。例えば、2001/12/23:01:02:03は、“0x20011223010203”と符号化される。

durationは、Clipの総再生時間をアライバルタイムクロックに基づいた時間／分／秒の単位で示した24ビットのフィールドである。このフィールドは、6個の数字を4ビットのBinary Coded Decimal (BCD)で符号化したものである。例えば、01:45:30は、“0x014530”と符号化される。

time_controlled_flag:のフラグは、AVストリームファイルの記録モードを示す。このtime_controlled_flagが1である場合、記録モードは、記録してからの時間経過に対してファイルサイズが比例するようにして記録されるモードであることを示し、次式に示す条件を満たさなければならない。

$$\text{TS_average_rate} * 192 / 188 * (\text{t} - \text{start_time}) - a \leq \text{size_clip}(\text{t})$$

$$<= \text{TS_average_rate} * 192 / 188 * (t - \text{start_time}) + \alpha$$

ここで、`TS_average_rate`は、AVストリームファイルのトランスポートストリームの平均ビットレートをbytes/secondの単位で表したものである。

また、上式において、`t`は、秒単位で表される時間を示し、`start_time`は、AVストリームファイルの最初のソースパケットが記録されたときの時刻であり、秒単位で表される。`size_clip(t)`は、時刻`t`におけるAVストリームファイルのサイズをバイト単位で表したものであり、例えば、`start_time`から時刻`t`までに10個のソースパケットが記録された場合、`size_clip(t)`は $10 * 192$ バイトである。`a`は、`TS_average_rate`に依存する定数である。

`time_controlled_flag`が0にセットされている場合、記録モードは、記録の時間経過とAVストリームのファイルサイズが比例するように制御していないことを示す。例えば、これは入力トランスポートストリームをトランスペアレント記録する場合である。

`TS_average_rate`は、`time_controlled_flag`が1にセットされている場合、この24ビットのフィールドは、上式で用いている`TS_average_rate`の値を示す。`time_controlled_flag`が0にセットされている場合、このフィールドは、何も意味を持たず、0にセットされなければならない。例えば、可変ビットレートのトランスポートストリームは、次に示す手順により符号化される。先ずトランスポートレートを`TS_recording_rate`の値にセットする。次に、ビデオストリームを可変ビットレートで符号化する。そして、ヌルパケットを使用しないことによって、間欠的にトランスポートパケットを符号化する。

`RSPN_arrival_time_discontinuity`の32ビットフィールドは、Bridge-Clip AV streamファイル上でアライバルタイムベースの不連続が発生する場所の相対アドレスである。`RSPN_arrival_time_discontinuity`は、ソースパケット番号を単位とする大きさであり、Bridge-Clip AV streamファイルの最初のソースパケットから`ClipInfo()`において定義される`offset_SPN`の値を初期値としてカウントされる。そのBridge-Clip AV streamファイルの中での絶対アドレスは、上述した

$$\text{SPN_xxx} = \text{RSPN_xxx} - \text{offset_SPN}$$

に基づいて算出される。

`reserved_for_system_use`の144ビットのフィールドは、システム用にリザーブされている。`is_format_identifier_valid`のフラグが1であるとき、`format_identifier`のフィールドが有効であることを示す。`is_original_network_ID_valid`のフラグが1である場合、`original_network_ID`のフィールドが有効であることを示す。`is_transport_stream_ID_valid`のフラグが1である場合、`transport_stream_ID`のフィールドが有効であることを示す。`is_servece_ID_valid`のフラグが1である場合、`servece_ID`のフィールドが有効であることを示す。

`is_country_code_valid`のフラグが1であるとき、`country_code`のフィールドが有効であることを示す。`format_identifier`の32ビットフィールドは、トランSPORTストリームの中で`registration deascriotor` (ISO/IEC13818-1で定義されている) が持つ`format_identifier`の値を示す。`original_network_ID`の16ビットフィールドは、トランSPORTストリームの中で定義されている`original_network_ID`の値を示す。`transport_stream_ID`の16ビットフィールドは、トランSPORTストリームの中で定義されている`transport_stream_ID`の値を示す。

`servece_ID`の16ビットフィールドは、トランSPORTストリームの中で定義されている`servece_ID`の値を示す。`country_code`の24ビットのフィールドは、ISO 3166によって定義されるカントリーコードを示す。それぞれのキャラクター文字は、ISO 8859-1で符号化される。例えば、日本は”JPN”と表され、”0x4A 0x50 0x4E”と符号化される。`stream_format_name`は、トランSPORTストリームのストリーム定義をしているフォーマット機関の名称を示すISO-646の16個のキャラクターコードである。このフィールドの中の無効なバイトは、値’0xFF’がセットされる。

`format_identifier`、`original_network_ID`、`transport_stream_ID`、`servece_ID`、`country_code`、及び`stream_format_name`は、トランSPORTストリームのサービスプロバイダを示すものであり、これにより、オーディオやビデオストリームの符号化制限、SI (サービスインフォメーション) の規格やオーディオビデオストリーム以外のプライベートデータストリームのストリーム定義を認識することができる。これらの情報は、デコーダが、そのストリームをデコードできるか否か、そしてデコードできる場合にデコード開始前にデコーダシステムの初期設

定を行うために用いることが可能である。

次に、STC_Infoについて説明する。ここでは、MPEG-2トランスポートストリームの中でSTCの不連続点（システムタイムベースの不連続点）を含まない時間区間をSTC_sequenceといい、Clipの中で、STC_sequenceは、STC_sequence_idの値によって特定される。図50A及び図50Bは、連続なSTC区間について説明する図である。同じSTC_sequenceの中で同じSTCの値は、決して現れない（但し、後述するように、Clipの最大時間長は制限されている）。したがって、同じSTC_sequenceの中で同じPTSの値もまた、決して現れない。AVストリームが、N(N>0)個のSTC不連続点を含む場合、Clipのシステムタイムベースは、(N+1)個のSTC_sequenceに分割される。

STC_Infoは、STCの不連続（システムタイムベースの不連続）が発生する場所のアドレスをストアする。図51を参照して説明するように、RSPN_STC_startが、そのアドレスを示し、最後のSTC_sequenceを除くk番目(k>=0)のSTC_sequenceは、k番目のRSPN_STC_startで参照されるソースパケットが到着した時刻から始まり、(k+1)番目のRSPN_STC_startで参照されるソースパケットが到着した時刻で終わる。最後のSTC_sequenceは、最後のRSPN_STC_startで参照されるソースパケットが到着した時刻から始まり、最後のソースパケットが到着した時刻で終了する。

図52は、STC_Infoのシンタクスを示す図である。図52に示したSTC_Infoのシンタクスについて説明すると、version_numberは、このSTC_Info()のバージョンナンバを示す4個のキャラクター文字である。version_numberは、ISO 646に従って、"0045"と符号化されなければならない。

lengthは、このlengthフィールドの直後からSTC_Info()の最後までのSTC_Info()のバイト数を示す32ビットの符号なし整数である。CPI()のCPI_typeがTU_map typeを示す場合、このlengthフィールドは0をセットしてもよい。CPI()のCPI_typeがEP_map typeを示す場合、num_of_STC_sequencesは1以上の値でなければならない。

num_of_STC_sequencesの8ビットの符号なし整数は、Clipの中でのSTC_sequenceの数を示す。この値は、このフィールドに続くfor-loopのループ回数を示す。

所定のSTC_sequenceに対応するSTC_sequence_idは、RSPN_STC_startを含むfor-loopの中で、そのSTC_sequenceに対応するRSPN_STC_startの現れる順番により定義されるものである。STC_sequence_idは、0から開始される。

RSPN_STC_startの32ビットフィールドは、AVストリームファイル上でSTC_sequenceが開始するアドレスを示す。RSPN_STC_startは、AVストリームファイルの中でシステムタイムベースの不連続点が発生するアドレスを示す。RSPN_STC_startは、AVストリームの中で新しいシステムタイムベースの最初のPCRを持つソースパケットの相対アドレスとしてもよい。RSPN_STC_startは、ソースパケット番号を単位とする大きさであり、AVストリームファイルの最初のソースパケットからClipInfo()において定義されるoffset_SPNの値を初期値としてカウントされる。そのAV streamファイルの中での絶対アドレスは、既に上述した

```
SPN_xxx = RSPN_xxx - offset_SPN
```

により算出される。

次に、図45に示したzzzzz.clipのシンタクス内のProgramInfoについて説明する。図53を参照しながら説明すると、ここでは、Clipの中で次の特徴を持つ時間区間をprogram_sequenceと呼ぶ。先ず、PCR_PIDの値が変わらない。次に、ビデオエレメンタリーストリームの数が変化しない。また、それぞれのビデオストリームについてのPIDの値とそのVideoCodingInfoによって定義される符号化情報が変化しない。更に、オーディオエレメンタリーストリームの数が変化しない。また、それぞれのオーディオストリームについてのPIDの値とそのAudioCodingInfoによって定義される符号化情報が変化しない。

program_sequenceは、同一の時刻において、ただ1つのシステムタイムベースを持つ。program_sequenceは、同一の時刻において、ただ1つのPMTを持つ。ProgramInfo()は、program_sequenceが開始する場所のアドレスをストアする。RSPN_program_sequence_startが、そのアドレスを示す。

図54は、ProgramInfoのシンタクスを示す図である。図54に示したProgramInfoのシンタクスを説明すると、version_numberは、このProgramInfo()のバージョンナンバを示す4個のキャラクター文字である。version_numberは、ISO 646に従って、"0045"と符号化されなければならない。

lengthは、このlengthフィールドの直後からProgramInfo () の最後までのProgramInfo () のバイト数を示す32ビットの符号なし整数である。CPI () のCPI_typeがTU_map_typeを示す場合、このlengthフィールドは0にセットされてもよい。CPI () のCPI_typeがEP_map_typeを示す場合、number_of_programsは1以上の値でなければならない。

number_of_program_sequencesの8ビットの符号なし整数は、Clipの中でのprogram_sequenceの数を示す。この値は、このフィールドに続くfor-loopのループ回数を示す。Clipの中でprogram_sequenceが変化しない場合、number_of_program_sequencesは1をセットされなければならない。RSPN_program_sequence_startの32ビットフィールドは、AVストリームファイル上でプログラムシーケンスが開始する場所の相対アドレスである。

RSPN_program_sequence_startは、ソースパケット番号を単位とする大きさであり、AVストリームファイルの最初のソースパケットからClipInfo () において定義されるoffset_SPNの値を初期値としてカウントされる。そのAVストリームファイルの中での絶対アドレスは、

SPN_xxx = RSPN_xxx - offset_SPN

により算出される。シンタクスのfor-loopの中でRSPN_program_sequence_start値は、昇順に現れなければならない。

PCR_PIDの16ビットフィールドは、そのprogram_sequenceに有効なPCRフィールドを含むトランスポートパケットのPIDを示す。number_of_videosの8ビットフィールドは、video_stream_PIDとVideoCodingInfo () を含むfor-loopのループ回数を示す。number_of_audiosの8ビットフィールドは、audio_stream_PIDとAudioCodingInfo () を含むfor-loopのループ回数を示す。video_stream_PIDの16ビットフィールドは、そのprogram_sequenceに有効なビデオストリームを含むトランスポートパケットのPIDを示す。このフィールドに続くVideoCodingInfo () は、そのvideo_stream_PIDで参照されるビデオストリームの内容を説明しなければならない。

audio_stream_PIDの16ビットフィールドは、そのprogram_sequenceに有効なオーディオストリームを含むトランスポートパケットのPIDを示す。このフィ

ールドに続くAudioCodingInfo () は、そのaudio_stream_PIDで参照されるビデオストリームの内容を説明しなければならない。

なお、シンタクスのfor-loopの中でvideo_stream_PIDの値の現れる順番は、そのprogram_sequenceに有効なPMTの中でビデオストリームのPIDが符号化されている順番に等しくなければならない。また、シンタクスのfor-loopの中でaudio_stream_PIDの値の現れる順番は、そのprogram_sequenceに有効なPMTの中でオーディオストリームのPIDが符号化されている順番に等しくなければならない。

図55は、図54に示したPrograminfoのシンタクス内のVideoCodingInfoのシンタクスを示す図である。図55に示したVideoCodingInfoのシンタクスを説明すると、video_formatの8ビットフィールドは、図56に示すように、ProgramInfo () の中のvideo_stream_PIDに対応するビデオフォーマットを示す。

frame_rateの8ビットフィールドは、図57に示すように、ProgramInfo () の中のvideo_stream_PIDに対応するビデオのフレームレートを示す。display_aspect_ratioの8ビットフィールドは、図58に示すように、ProgramInfo () の中のvideo_stream_PIDに対応するビデオの表示アスペクト比を示す。

図59は、図54に示したPrograminfoのシンタクス内のAudioCodingInfoのシンタクスを示す図である。図59に示したAudioCodingInfoのシンタクスを説明すると、audio_codingの8ビットフィールドは、図60に示すように、ProgramInfo () の中のaudio_stream_PIDに対応するオーディオの符号化方法を示す。

audio_component_typeの8ビットフィールドは、図61に示すように、ProgramInfo () の中のaudio_stream_PIDに対応するオーディオのコンポーネントタイプを示す。sampling_frequencyの8ビットフィールドは、図62に示すように、ProgramInfo () の中のaudio_stream_PIDに対応するオーディオのサンプリング周波数を示す。

次に、図45に示したzzzzz.clipのシンタクス内のCPI (Characteristic Point Information)について説明する。CPIは、AVストリームの中の時間情報とそのファイルの中のアドレスとを関連付けるためにある。CPIには2つのタイプがあり、それらはEP_mapとTU_mapである。図63に示すように、CPI () の

中のCPI_typeがEP_map typeの場合、そのCPI()はEP_mapを含む。図64に示すように、CPI()の中のCPI_typeがTU_map typeの場合、そのCPI()はTU_mapを含む。1つのAVストリームは、1つのEP_map又は1つのTU_mapを持つ。AVストリームがS E S FトランSPORTストリームの場合、それに対応するClipはEP_mapを持たなければならない。

図65は、CPIのシンタクスを示す図である。図65に示したCPIのシンタクスを説明すると、version_numberは、このCPI()のバージョンナンバを示す4個のキャラクター文字である。version_numberは、ISO 646に従って、"0045"と符号化されなければならない。lengthは、このlengthフィールドの直後からCPI()の最後までのCPI()のバイト数を示す32ビットの符号なし整数である。CPI_typeは、図66に示すように、1ビットのフラグであり、ClipのCPIのタイプを表す。

次に、図65に示したCPIのシンタクス内のEP_mapについて説明する。EP_mapには、2つのタイプがあり、それはビデオストリーム用のEP_mapとオーディオストリーム用のEP_mapである。EP_mapの中のEP_map_typeが、EP_mapのタイプを区別する。Clipが1つ以上のビデオストリームを含む場合、ビデオストリーム用のEP_mapが使用されなければならない。Clipがビデオストリームを含まず、1つ以上のオーディオストリームを含む場合、オーディオストリーム用のEP_mapが使用されなければならない。

ビデオストリーム用のEP_mapについて図67を参照して説明する。ビデオストリーム用のEP_mapは、stream_PID、PTS_EP_start、及び、RSPN_EP_startというデータを持つ。stream_PIDは、ビデオストリームを伝送するトランSPORTパケットのPIDを示す。PTS_EP_startは、ビデオストリームのシーケンスヘッダから始めるアクセスユニットのPTSを示す。RSPN_EP_startは、AVストリームの中でPTS_EP_startにより参照されるアクセスユニットの第1バイト目を含むソースポケットのアドレスを示す。

EP_map_for_one_stream_PID()と呼ばれるサブテーブルは、同じPIDを持つトランSPORTパケットによって伝送されるビデオストリーム毎に作られる。Clipの中に複数のビデオストリームが存在する場合、EP_mapは複数のEP_map_for_o

ne_stream_PID () を含んでもよい。

オーディオストリーム用のEP_mapは、stream_PID、PTS_EP_start、及びRSPN_EP_startというデータを持つ。stream_PIDは、オーディオストリームを伝送するトランスポートパケットのPIDを示す。PTS_EP_startは、オーディオストリームのアクセスユニットのPTSを示す。RSPN_EP_startは、AVストリームの中でPTS_EP_startで参照されるアクセスユニットの第1バイト目を含むソースポケットのアドレスを示す。

EP_map_for_one_stream_PID () と呼ばれるサブテーブルは、同じPIDを持つトランスポートパケットによって伝送されるオーディオストリーム毎に作られる。Clipの中に複数のオーディオストリームが存在する場合、EP_mapは複数のEP_map_for_one_stream_PID () を含んでもよい。

EP_mapとSTC_Infoの関係を説明すると、1つのEP_map_for_one_stream_PID () は、STCの不連続点に関係なく1つのテーブルに作られる。RSPN_EP_startの値とSTC_Info () において定義されるRSPN_STC_startの値を比較することにより、それぞれのSTC_sequenceに属するEP_mapのデータの境界が分かる（図68を参照）。EP_mapは、同じPIDで伝送される連続したストリームの範囲に対して、1つのEP_map_for_one_stream_PIDを持たねばならない。図69に示したような場合、program#1とprogram#3は、同じビデオPIDを持つが、データ範囲が連続していないので、それぞれのプログラム毎にEP_map_for_one_stream_PIDを持たねばならない。

図70は、EP_mapのシンタクスを示す図である。図70に示したEP_mapのシンタクスを説明すると、EP_typeは、4ビットのフィールドであり、図71に示すように、EP_mapのエントリポイントタイプを示す。EP_typeは、このフィールドに続くデータフィールドのセマンティクスを示す。Clipが1つ以上のビデオストリームを含む場合、EP_typeは0 ('video') にセットされなければならない。又は、Clipがビデオストリームを含まず、1つ以上のオーディオストリームを含む場合、EP_typeは1 ('audio') にセットされなければならない。

number_of_stream_PIDsの16ビットのフィールドは、EP_map () の中のnumber_of_stream_PIDsを変数を持つfor-loopのループ回数を示す。stream_PID (k) の

16ビットのフィールドは、EP_map_for_one_stream_PID (num_EP_entries (k))によって参照されるk番目のエレメンタリーストリーム（ビデオ又はオーディオストリーム）を伝送するトランスポートパケットのPIDを示す。EP_typeが0 ('video')に等しい場合、そのエレメンタリーストリームはビデオストリームでなければならない。また、EP_typeが1 ('audio')に等しい場合、そのエレメンタリーストリームはオーディオストリームでなければならない。

num_EP_entries (k) の16ビットのフィールドは、EP_map_for_one_stream_PID (num_EP_entries (k))によって参照されるnum_EP_entries (k)を示す。EP_map_for_one_stream_PID_Start_address (k)：この32ビットのフィールドは、EP_map ()の中でEP_map_for_one_stream_PID (num_EP_entries (k))が始まる相対バイト位置を示す。この値は、EP_map ()の第1バイト目からの大きさで示される。

padding_wordは、EP_map ()のシンタクスに従って挿入されなければならない。XとYは、0又は任意の正の整数でなければならない。それぞれのパディングワードは、任意の値を取ってもよい。

図72は、EP_map_for_one_stream_PIDのシンタクスを示す図である。図72に示したEP_map_for_one_stream_PIDのシンタクスを説明すると、PTS_EP_startの32ビットのフィールドのセマンティクスは、EP_map ()において定義されるEP_typeにより異なる。EP_typeが0 ('video')に等しい場合、このフィールドは、ビデオストリームのシーケンスヘッダで始まるアクセスユニットの33ビット精度のPTSの上位32ビットを持つ。EP_typeが1 ('audio')に等しい場合、このフィールドは、オーディオストリームのアクセスユニットの33ビット精度のPTSの上位32ビットを持つ。

RSPN_EP_startの32ビットのフィールドのセマンティクスは、EP_map ()において定義されるEP_typeにより異なる。EP_typeが0 ('video')に等しい場合、このフィールドは、AVストリームの中でPTS_EP_startにより参照されるアクセスユニットのシーケンスヘッダの第1バイト目を含むソースパケットの相対アドレスを示す。又は、EP_typeが1 ('audio')に等しい場合、このフィールドは、AVストリームの中でPTS_EP_startにより参照されるアクセスユニットのオーディ

オフレームの第一バイト目を含むソースポケットの相対アドレスを示す。

RSPN_EP_startは、ソースパケット番号を単位とする大きさであり、AVストリームファイルの最初のソースパケットからClipInfo()において定義されるoffset_SPNの値を初期値としてカウントされる。そのAVストリームファイルの中での絶対アドレスは、

$$\text{SPN_xxx} = \text{RSPN_xxx} - \text{offset_SPN}$$

により算出される。シンタクスのfor-loopの中でRSPN_EP_startの値は、昇順に現れなければならない。

次に、TU_mapについて、図73を参照して説明する。TU_mapは、ソースパケットのアライバルタイムクロック（到着時刻ベースの時計）に基づいて、1つの時間軸を作る。その時間軸は、TU_map_time_axisと呼ばれる。TU_map_time_axisの原点は、TU_map()の中のoffset_timeによって示される。TU_map_time_axisは、offset_timeから一定の単位に分割される。その単位を、time_unitという。

AVストリームの中の各々のtime_unitの中で、最初の完全な形のソースパケットのAVストリームファイル上のアドレスが、TU_mapにストアされる。これらのアドレスを、RSPN_time_unit_startという。TU_map_time_axis上において、k ($k \geq 0$) 番目のtime_unitが始まる時刻は、TU_start_time(k)と呼ばれる。この値は次式に基づいて算出される。

$$\text{TU_start_time}(k) = \text{offset_time} + k * \text{time_unit_size}$$

TU_start_time(k)は、45 kHzの精度を持つ。

図75は、TU_mapのシンタクスを示す図である。図75に示したTU_mapのシンタクスを説明すると、offset_timeの32bit長のフィールドは、TU_map_time_axisに対するオフセットタイムを与える。この値は、Clipの中の最初のtime_unitに対するオフセット時刻を示す。offset_timeは、27MHz精度のアライバルタイムクロックから導き出される45kHzクロックを単位とする大きさである。AVストリームが新しいClipとして記録される場合、offset_timeは0にセットされなければならない。

time_unit_sizeの32ビットフィールドは、time_unitの大きさを与えるものであり、それは27MHz精度のアライバルタイムクロックから導き出される45

k Hz クロックを単位とする大きさである。time_unit_sizeは、1秒以下 (time_unit_size<=45000) にすることがよい。number_of_time_unit_entriesの32ビットフィールドは、TU_map () の中にストアされているtime_unitのエントリ数を示す。

RSPN_time_unit_startの32ビットフィールドは、AVストリームの中でそれそのtime_unitが開始する場所の相対アドレスを示す。RSPN_time_unit_startは、ソースパケット番号を単位とする大きさであり、AV streamファイルの最初のソースパケットからClipInfo () において定義されるoffset_SPNの値を初期値としてカウントされる。そのAV streamファイルの中での絶対アドレスは、

SPN_xxx = RSPN_xxx - offset_SPN

により算出される。シンタクスのfor-loopの中でRSPN_time_unit_startの値は、昇順に現れなければならない。(k + 1) 番目のtime_unitの中にソースパケットが何もない場合、(k + 1) 番目のRSPN_time_unit_startは、k 番目のRSPN_time_unit_startと等しくなければならない。

図45に示したzzzz.clipのシンタクス内のClipMarkについて説明する。ClipMarkは、クリップについてのマーク情報であり、ClipMarkの中にストアされる。このマークは、記録器（記録再生装置1）によってセットされるものであり、ユーザによってセットされるものではない。

図75は、ClipMarkのシンタクスを示す図である。図75に示したClipMarkのシンタクスを説明すると、version_numberは、このClipMark () のバージョンナンバを示す4個のキャラクター文字である。version_numberは、ISO 646に従って、“0045”と符号化されなければならない。

lengthは、このlengthフィールドの直後からClipMark () の最後までのClipMark () のバイト数を示す32ビットの符号なし整数である。number_of_Clip_marksは、ClipMarkの中にストアされているマークの個数を示す16ビットの符号なし整数であり、number_of_Clip_marksは、0であってもよい。mark_typeは、マークのタイプを示す8ビットのフィールドであり、図76に示すテーブルに従つて符号化される。

mark_time_stampは、32ビットフィールドであり、マークが指定されたポイント

トを示すタイムスタンプをストアする。mark_time_stampのセマンティクスは、図77に示すように、PlayList()の中のCPI_typeにより異なる。

STC_sequence_idは、CPI()の中のCPI_typeがEP_map_typeを示す場合、この8ビットのフィールドは、マークが置かれているところのSTC連続区間のSTC_sequence_idを示す。CPI()の中のCPI_typeがTU_map_typeを示す場合、この8ビットのフィールドは何も意味を持たず、0にセットされる。character_setの8ビットのフィールドは、mark_nameフィールドに符号化されているキャラクター文字の符号化方法を示す。その符号化方法は、図19に示される値に対応する。

name_lengthの8ビットフィールドは、Mark_nameフィールドの中に示されるマーク名のバイト長を示す。mark_nameのフィールドは、マークの名称を示す。このフィールドの中の左からname_length数のバイト数が、有効なキャラクター文字であり、それはマークの名称を示す。mark_nameフィールドの中で、それら有効なキャラクター文字の後の値は、どんな値が入っていてもよい。

ref_thumbnail_indexのフィールドは、マークに付加されるサムネイル画像の情報を示す。ref_thumbnail_indexフィールドが、0xFFFFでない値の場合、そのマークにはサムネイル画像が付加されており、そのサムネイル画像は、mark.thmbファイルの中にストアされている。その画像は、mark.thmbファイルの中でref_thumbnail_indexの値を用いて参照される。ref_thumbnail_indexフィールドが、0xFFFFである場合、そのマークにはサムネイル画像が付加されていない。

MakersPrivateDataについては、図22を参照して既に説明したので、その説明は省略する。

次に、サムネイルインフォメーション(Thumbnail Information)について説明する。サムネイル画像は、menu.thmbファイル又はmark.thmbファイルにストアされる。これらのファイルは同じシンタクス構造であり、ただ1つのThumbnail()を持つ。menu.thmbファイルは、メニュー サムネイル画像、すなわちVolumeを代表する画像、及び、それぞれのPlayListを代表する画像をストアする。全てのメニュー サムネイルは、ただ1つのmenu.thmbファイルにストアされる。

mark.thmbファイルは、マーク サムネイル画像、すなわちマーク点を表すピクチャをストアする。全てのPlayList及びClipに対する全てのマーク サムネイルは、

ただ1つのmark.thmbファイルにストアされる。サムネイルは頻繁に追加、削除されるので、追加操作と部分削除の操作は容易に高速に実行できなければならない。この理由のため、Thumbnail () はロック構造を有する。画像のデータはいくつかの部分に分割され、各部分は1つのtn_blockに格納される。1つの画像データは連続したtn_blockに格納される。tn_blockの列には、使用されていないtn_blockが存在してもよい。1つのサムネイル画像のバイト長は可変である。

図78は、menu.thmbとmark.thmbのシンタクスを示す図であり、図79は、図78に示したmenu.thmbとmark.thmbのシンタクス内のThumbnailのシンタクスを示す図である。図79に示したThumbnailのシンタクスについて説明すると、version_numberは、このThumbnail () のバージョンナンバを示す4個のキャラクター文字である。version_numberは、ISO 646に従って、"0045"と符号化されなければならない。

lengthは、このlengthフィールドの直後からThumbnail () の最後までのMakersPrivateData () のバイト数を示す32ビットの符号なし整数である。tn_blocks_start_addressは、Thumbnail () の先頭のバイトからの相対バイト数を単位として、最初のtn_blockの先頭バイトアドレスを示す32ビットの符号なし整数である。相対バイト数は0からカウントされる。number_of_thumbnailsは、Thumbnail () の中に含まれているサムネイル画像のエントリ数を与える16ビットの符号なし整数である。

tn_block_sizeは、1024バイトを単位として、1つのtn_blockの大きさを与える16ビットの符号なし整数である。例えば、tn_block_size=1ならば、それは1つのtn_blockの大きさが1024バイトであることを示す。number_of_tn_blocksは、このThumbnail () 中のtn_blockのエントリ数を表す116ビットの符号なし整数である。thumbnail_indexは、このthumbnail_indexフィールドから始まるforループ一回分のサムネイル情報で表されるサムネイル画像のインデックス番号を表す16ビットの符号なし整数である。thumbnail_indexとして、0xFFFFという値を使用してはならない。thumbnail_indexはUIAppInfoVolume () 、UIAppInfoPlayList () 、PlayListMark () 、及びClipMark () の中のref_thumbnail_indexによって参照される。

`thumbnail_picture_format`は、サムネイル画像のピクチャフォーマットを表す8ビットの符号なし整数で、図80に示すような値をとる。表中のDCFとPNGは”`menu.thmb`”内でのみ許される。マークサムネイルは、値”0x00”（MPEG-2 Video I-picture）をとらなければならない。

`picture_data_size`は、サムネイル画像のバイト長をバイト単位で示す32ビットの符号なし整数である。`start_tn_block_number`は、サムネイル画像のデータが始まる`tn_block`の`tn_block`番号を表す16ビットの符号なし整数である。サムネイル画像データの先頭は、`tb_block`の先頭と一致していなければならない。`tn_block`番号は、0から始まり、`tn_block`のfor-ループ中の変数kの値に関係する。

`x_picture_length`は、サムネイル画像のフレーム画枠の水平方向のピクセル数を表す16ビットの符号なし整数である。`y_picture_length`は、サムネイル画像のフレーム画枠の垂直方向のピクセル数を表す16ビットの符号なし整数である。`tn_block`は、サムネイル画像がストアされる領域である。`Thumbnail()`の中の全ての`tn_block`は、同じサイズ（固定長）であり、その大きさは`tn_block_size`によって定義される。

図81A及び図81Bは、サムネイル画像データがどのように`tn_block`に格納されるかを模式的に表した図である。図81A及び図81Bのように、各サムネイル画像データは`tn_block`の先頭から始まり、1`tn_block`を超える大きさの場合は、連続する次の`tn_block`を使用してストアされる。このようにすることにより、可変長であるピクチャデータが、固定長のデータとして管理することが可能となり、削除といった編集に対して簡便な処理により対応することができるようになる。

次に、AVストリームファイルについて説明する。AVストリームファイルは、“M2TS”ディレクトリ（図14）にストアされる。AVストリームファイルには、2つのタイプがあり、それらは、Clip AVストリームとBridge-Clip AVストリームファイルである。両方のAVストリーム共に、これ以降で定義されるDVR MPEG-2トランSPORTストリームファイルの構造でなければならない。

先ず、DVR MPEG-2トランSPORTストリームについて説明する。DVR MPEG-2トランSPORTストリームの構造は、図82に示すようにな

っている。AVストリームファイルは、DVR MPEG2トランSPORTストリームの構造を持つ。DVR MPEG2トランSPORTストリームは、整数個のAligned unitから構成される。Aligned unitの大きさは、6144バイト(2048*3バイト)である。Aligned unitは、ソースパケットの第1バイト目から始まる。ソースパケットは、192バイト長である。1つのソースパケットは、TP_extra_headerとトランSPORTパケットからなる。TP_extra_headerは、4バイト長であり、またトランSPORTパケットは、188バイト長である。

1つのAligned unitは、32個のソースパケットからなる。DVR MPEG2トランSPORTストリームの中の最後のAligned unitも、また32個のソースパケットからなる。よって、DVR MPEG2トランSPORTストリームは、Aligned unitの境界で終端する。ディスクに記録される入力トランSPORTストリームのトランSPORTパケットの数が32の倍数でないとき、ヌルパケット(PID=0x1FFFのトランSPORTパケット)を持ったソースパケットを最後のAligned unitに使用しなければならない。ファイルシステムは、DVR MPEG2トランSPORTストリームに余分な情報を付加してはならない。

図83に、DVR MPEG-2トランSPORTストリームのレコーダモデルを示す。図83に示したレコーダは、レコーディングプロセスを規定するための概念上のモデルである。DVR MPEG-2トランSPORTストリームは、このモデルに従う。

MPEG-2トランSPORTストリームの入力タイミングについて説明する。入力MPEG2トランSPORTストリームは、フルトランSPORTストリーム又はパーシャルトランSPORTストリームである。入力されるMPEG2トランSPORTストリームは、ISO/IEC13818-1又はISO/IEC13818-9に従っていなければならない。MPEG2トランSPORTストリームのi番目のバイトは、T-STD(ISO/IEC13818-1で規定されるTransport stream system target decoder)とソースパケッタイザーへ、時刻(i)に同時に入力される。Rpkは、トランSPORTパケットの入力レートの瞬時的な最大値である。

27MHz PLL52は、27MHzクロックの周波数を発生する。27MHzクロ

ックの周波数は、M P E G - 2 トランスポートストリームのP C R (Program Clock Reference) の値にロックされる。arrival time clock counter 5 3は、27 M H z の周波数のパルスをカウントするバイナリーカウンターである。Arrival_time_clock (i) は、時刻t (i) におけるArrival time clock counterのカウント値である。

source packetizer 5 4は、全てのトランスポートパケットにTP_extra_header を付加し、ソースパケットを作る。Arrival_time_stampは、トランスポートパケットの第1バイト目がT-S T Dとソースパケットタイマーの両方へ到着する時刻を表す。Arrival_time_stamp (k) は、次式で示されるようにArrival_time_clock (k) のサンプル値であり、ここで、kはトランスポートパケットの第1バイト目を示す。

$$\text{arrival_time_stamp (k)} = \text{arrival_time_clock (k)} \% 230$$

2つの連続して入力されるトランスポートパケットの時間間隔が、230/2700000 00秒(約40秒)以上になる場合、その2つのトランスポートパケットのarrival_time_stampの差分は、230/27000000秒になるようにセットされるべきである。レコーダは、そのようになる場合に備えてある。

smoothing buffer 5 5は、入力トランスポートストリームのビットレートをスマージングする。スマージングバッファは、オーバーフロウしてはならない。Rmaxは、スマージングバッファが空でないときのスマージングバッファからのソースパケットの出力ビットレートである。スマージングバッファが空であるとき、スマージングバッファからの出力ビットレートは0である。

次に、D V R M P E G - 2 トランスポートストリームのレコーダモデルのパラメータについて説明する。Rmaxという値は、A Vストリームファイルに対応するClipInfo ()において定義されるTS_recording_rateによって与えられる。この値は、次式により算出される。

$$R_{\max} = TS_recording_rate * 192/188$$

TS_recording_rateの値は、bytes/secondを単位とする大きさである。

入力トランスポートストリームがS E S Fトランスポートストリームの場合、Rpkは、A Vストリームファイルに対応するClipInfo ()において定義されるTS_

recording_rateに等しくなければならない。入力トランSPORTストリームがS E S FトランSPORTストリームでない場合、この値はM P E G - 2 transport streamのデスクリプター、例えばmaximum_bitrate_descriptorやpartial_transport_stream_descriptor等において定義される値を参照してもよい。

smoothing buffer sizeは、入力トランSPORTストリームがS E S FトランSPORTストリームの場合、スムージングバッファの大きさは0である。入力トランSPORTストリームがS E S FトランSPORTストリームでない場合、スムージングバッファの大きさはM P E G - 2 transport streamのデスクリプター、例えばsmoothing_buffer_descriptor、short_smoothing_buffer_descriptor、partial_transport_stream_descriptor等において定義される値を参照してもよい。

記録機（レコーダ）及び再生機（プレーヤ）は、十分なサイズのバッファを用意しなければならない。デフォルトのバッファサイズは、1536 bytesである。

次に、D V R M P E G - 2トランSPORTストリームのプレーヤモデルについて説明する。図84は、D V R M P E G - 2トランSPORTストリームのプレーヤモデルを示す図である。これは、再生プロセスを規定するための概念上のモデルである。D V R M P E G - 2トランSPORTストリームは、このモデルに従う。

27MHz X-tal 6 1は、27MHzの周波数を発生する。27MHz周波数の誤差範囲は、+/-30 ppm (27000000 +/- 810 Hz)でなければならない。arrival time clock counter 6 2は、27MHzの周波数のパルスをカウントするバイナリーカウンターである。Arrival_time_clock (i)は、時刻t (i)におけるArrival time clock counterのカウント値である。

smoothing buffer 6 4において、Rmaxは、スムージングバッファがフルでないときのスムージングバッファへのソースパケットの入力ビットレートである。スムージングバッファがフルであるとき、スムージングバッファへの入力ビットレートは0である。

M P E G - 2トランSPORTストリームの出力タイミングを説明すると、現在のソースパケットのarrival_time_stampがarrival_time_clock (i)のLSB 30ビットの値と等しいとき、そのソースパケットのトランSPORTパケットは、スムージングバッファから引き抜かれる。Rpkは、トランSPORTパケットレートの瞬

時的な最大値である。スムージングバッファは、アンダーフロウしてはならない。

DVR MPEG-2トランSPORTストリームのプレーヤモデルのパラメータについては、上述したDVR MPEG-2トランSPORTストリームのレコーダモデルのパラメータと同一である。

図85は、Source packetのシンタクスを示す図である。`transport_packet()`は、ISO/IEC 13818-1で規定されるMPEG-2トランSPORTパケットである。図85に示したSource packetのシンタクス内の`TP_Extra_header`のシンタクスを図86に示す。図86に示した`TP_Extra_header`のシンタクスについて説明すると、`copy_permission_indicator`は、トランSPORTパケットのペイロードのコピー制限を表す整数である。コピー制限は、`copy free`、`no more copy`、`copy once`、又は`copy prohibited`とすることができる。図87は、`copy_permission_indicator`の値と、それらによって指定されるモードの関係を示す。

`copy_permission_indicator`は、全てのトランSPORTパケットに付加される。IEEE1394ディジタルインタフェースを使用して入力トランSPORTストリームを記録する場合、`copy_permission_indicator`の値は、`IEEE1394 isochronous packet header`の中の`EMI` (Encryption Mode Indicator) の値に関連付けてもよい。IEEE1394ディジタルインタフェースを使用しないで入力トランSPORTストリームを記録する場合、`copy_permission_indicator`の値は、トランSPORTパケットの中に埋め込まれた`CCI`の値に関連付けてもよい。アナログ信号入力をセルフエンコードする場合、`copy_permission_indicator`の値は、アナログ信号のCGMS-Aの値に関連付けてもよい。

`arrival_time_stamp`は、次式

$$\text{arrival_time_stamp}(k) = \text{arrival_time_clock}(k) \% 230$$

において、`arrival_time_stamp`によって指定される値を持つ整数値である。

Clip AVストリームの定義をすると、Clip AVストリームは、上述したような定義がされるDVR MPEG-2トランSPORTストリームの構造を持たねばならない。`arrival_time_clock(i)`は、Clip AVストリームの中で連続して増加しなければならない。Clip AVストリームの中にシステムタイムベース (STCベース) の不連続点が存在したとしても、そのClip AVストリームの`arrival`

_time_clock (i) は、連続して増加しなければならない。

Clip AVストリームの中の開始と終了の間のarrival_time_clock (i) の差分の最大値は、26時間でなければならない。この制限は、MPEG2トランSPORTストリームの中にシステムタイムベース(STCベース)の不連続点が存在しない場合に、Clip AVストリームの中で同じ値のPTS(Presentation Time Stamp)が決して現れないことを保証する。MPEG2システムズ規格は、PTSのラップアラウンド周期を233/90000秒(約26.5時間)と規定している。

Bridge-Clip AVストリームの定義をすると、Bridge-Clip AVストリームは、上述したような定義がされるDVR MPEG-2トランSPORTストリームの構造を持たねばならない。Bridge-Clip AVストリームは、1つのアライバルタイムベースの不連続点を含まなければならない。アライバルタイムベースの不連続点の前後のトランSPORTストリームは、後述する符号化の制限に従わなければならず、且つ後述するDVR-STDに従わなければならない。

本例においては、編集におけるPlayItem間のビデオとオーディオのシームレス接続をサポートする。PlayItem間をシームレス接続にすることは、プレーヤ/レコーダに”データの連続供給”と”シームレスな復号処理”を保証する。”データの連続供給”とは、ファイルシステムが、デコーダにバッファのアンダーフロウを起こさせることのないように必要なビットレートでデータを供給することを保証できることである。データのリアルタイム性を保証して、データをディスクから読み出すことができるよう、データが十分な大きさの連続したブロック単位でストアされるようにする。

”シームレスな復号処理”とは、プレーヤが、デコーダの再生出力にポーズやギャップを起こさせることなく、ディスクに記録されたオーディオビデオデータを表示できることである。

シームレス接続されているPlayItemが参照するAVストリームについて説明する。先行するPlayItemと現在のPlayItemの接続が、シームレス表示できるように保証されているかどうかは、現在のPlayItemにおいて定義されているconnection_conditionフィールドから判断することができる。PlayItem間のシームレス接続は、Bridge-Clipを使用する方法と使用しない方法がある。

図88は、Bridge-Clipを使用する場合の先行するPlayItemと現在のPlayItemの関係を示している。図88においては、プレーヤが読み出すストリームデータが、影を付けて示されている。図88に示したTS1は、Clip1 (Clip AVストリーム) の影を付けられたストリームデータとBridge-ClipのRSPN_arrival_time_discontinuityより前の影を付けられたストリームデータからなる。

TS1のClip1の影を付けられたストリームデータは、先行するPlayItemのIN_time (図88においてIN_time1で図示されている) に対応するプレゼンテーションユニットを復号するために必要なストリームのアドレスから、RSPN_exit_from_previous_Clipで参照されるソースパケットまでのストリームデータである。TS1に含まれるBridge-ClipのRSPN_arrival_time_discontinuityより前の影を付けられたストリームデータは、Bridge-Clipの最初のソースパケットから、RSPN_arrival_time_discontinuityで参照されるソースパケットの直前のソースパケットまでのストリームデータである。

また、図88におけるTS2は、Clip2 (Clip AVストリーム) の影を付けられたストリームデータとBridge-ClipのRSPN_arrival_time_discontinuity以後の影を付けられたストリームデータからなる。TS2に含まれるBridge-ClipのRSPN_arrival_time_discontinuity以後の影を付けられたストリームデータは、RSPN_arrival_time_discontinuityで参照されるソースパケットから、Bridge-Clipの最後のソースパケットまでのストリームデータである。TS2のClip2の影を付けられたストリームデータは、RSPN_enter_to_current_Clipで参照されるソースパケットから、現在のPlayItemのOUT_time (図88においてOUT_time2で図示されている) に対応するプレゼンテーションユニットを復号するために必要なストリームのアドレスまでのストリームデータである。

図89は、Bridge-Clipを使用しない場合の先行するPlayItemと現在のPlayItemの関係を示している。この場合、プレーヤが読み出すストリームデータは、影を付けて示されている。図89におけるTS1は、Clip1 (Clip AVストリーム) の影を付けられたストリームデータからなる。TS1のClip1の影を付けられたストリームデータは、先行するPlayItemのIN_time (図89においてIN_time1で図示されている) に対応するプレゼンテーションユニットを復号するために必要なス

トリームのアドレスから始まり、Clip1の最後のソースパケットまでのデータである。また、図8.9におけるTS2は、Clip2 (Clip AVストリーム) の影を付けられたストリームデータからなる。

TS2のClip2の影を付けられたストリームデータは、Clip2の最初のソースパケットから始まり、現在のPlayItemのOUT_time (図8.9においてOUT_time2で図示されている) に対応するプレゼンテーションユニットを復号するために必要なストリームのアドレスまでのストリームデータである。

図8.8と図8.9において、TS1とTS2は、ソースパケットの連続したストリームである。次に、TS1とTS2のストリーム規定と、それらの間の接続条件について考える。先ず、シームレス接続のための符号化制限について考える。トランスポートストリームの符号化構造の制限として、先ず、TS1とTS2の中に含まれるプログラムの数は、1でなければならない。TS1とTS2の中に含まれるビデオストリームの数は、1でなければならない。TS1とTS2の中に含まれるオーディオストリームの数は、2以下でなければならない。TS1とTS2の中に含まれるオーディオストリームの数は、等しくなければならない。TS1及び/又はTS2の中に、上記以外のエレメンタリーストリーム又はプライベートストリームが含まれていてもよい。

ビデオビットストリームの制限について説明する。図9.0は、ピクチャの表示順序で示すシームレス接続の例を示す図である。接続点においてビデオストリームをシームレスに表示できるためには、OUT_time1 (Clip1のOUT_time) の後とIN_time2 (Clip2のIN_time) の前に表示される不必要なピクチャは、接続点付近のClipの部分的なストリームを再エンコードするプロセスにより、除去されなければならない。

図9.0に示したような場合において、BridgeSequenceを使用してシームレス接続を実現する例を、図9.1に示す。RSPN_arrival_time_discontinuityより前のBridge-Clipのビデオストリームは、図9.0のClip1のOUT_time1に対応するピクチャまでの符号化ビデオストリームからなる。そして、そのビデオストリームは先行するClip1のビデオストリームに接続され、1つの連續でMPEG2規格に従ったエレメンタリーストリームとなるように再エンコードされている。

同様にして、RSPN_arrival_time_discontinuity以後のBridge-Clipのビデオストリームは、図90のClip2のIN_time2に対応するピクチャ以後の符号化ビデオストリームからなる。そして、そのビデオストリームは、正しくデコード開始することができて、これに続くClip2のビデオストリームに接続され、1つの連続でMPEG2規格に従ったエレメンタリーストリームとなるように再エンコードされている。Bridge-Clipを作るためには、一般に、数枚のピクチャは再エンコードしなければならず、それ以外のピクチャはオリジナルのClipからコピーすることができる。

図90に示した例の場合BridgeSequenceを使用しないでシームレス接続を実現する例を図92に示す。Clip1のビデオストリームは、図90のOUT_time1に対応するピクチャまでの符号化ビデオストリームからなり、それは、1つの連続でMPEG2規格に従ったエレメンタリーストリームとなるように再エンコードされている。同様にして、Clip2のビデオストリームは、図90のClip2のIN_time2に対応するピクチャ以後の符号化ビデオストリームからなり、それは、1つの連続でMPEG2規格に従ったエレメンタリーストリームとなるように再エンコードされている。

ビデオストリームの符号化制限について説明すると、先ず、TS1とTS2のビデオストリームのフレームレートは、等しくなければならない。TS1のビデオストリームは、sequence_end_codeで終端しなければならない。TS2のビデオストリームは、Sequence Header、GOP Header、そしてIピクチャで開始しなければならない。TS2のビデオストリームは、クローズドGOPで開始しなければならない。

ビットストリームの中で定義されるビデオプレゼンテーションユニット（フレーム又はフィールド）は、接続点を挟んで連続でなければならない。接続点において、フレーム又はフィールドのギャップがあつてはならない。接続点において、トップ-ボトムのフィールドシーケンスは連続でなければならない。3-2ブルダウンを使用するエンコードの場合は、"top_field_first" 及び "repeat_first_field" フラグを書き換える必要があるかもしれない、又はフィールドギャップの発生を防ぐために局所的に再エンコードするようにしてもよい。

オーディオビットストリームの符号化制限について説明すると、TS1とTS2のオーディオのサンプリング周波数は、同じでなければならない。TS1とTS2のオーディオの符号化方法（例、MPEG1レイヤ2、AC-3、SSEFLPCM、AAC）は、同じでなければならない。

次に、MPEG-2トランスポートストリームの符号化制限について説明すると、TS1のオーディオストリームの最後のオーディオフレームは、TS1の最後の表示ピクチャの表示終了時に等しい表示時刻を持つオーディオサンプルを含んでいなければならない。TS2のオーディオストリームの最初のオーディオフレームは、TS2の最初の表示ピクチャの表示開始時に等しい表示時刻を持つオーディオサンプルを含んでいなければならない。

接続点において、オーディオプレゼンテーションユニットのシーケンスにギャップがあつてはならない。図93に示すように、2オーディオフレーム区間未満のオーディオプレゼンテーションユニットの長さで定義されるオーバーラップがあつてもよい。TS2のエレメンタリーストリームを伝送する最初のパケットは、ビデオパケットでなければならない。接続点におけるトランスポートストリームは、後述するDVR-STDに従わなくてはならない。

Clip及びBridge-Clipの制限について説明すると、TS1とTS2は、それの中にアライバルタイムベースの不連続点を含んではならない。

以下の制限は、Bridge-Clipを使用する場合にのみ適用される。TS1の最後のソースパケットとTS2の最初のソースパケットの接続点においてのみ、Bridge-Clip AVストリームは、ただ1つのアライバルタイムベースの不連続点を持つ。ClipInfo()において定義されるRSPN_arrival_time_discontinuityが、その不連続点のアドレスを示し、それはTS2の最初のソースパケットを参照するアドレスを示さなければならない。

BridgeSequenceInfo()において定義されるRSPN_exit_from_previous_Clipによって参照されるソースパケットは、Clip1の中のどのソースパケットでもよい。それは、Aligned unitの境界である必要はない。BridgeSequenceInfo()において定義されるRSPN_enter_to_current_Clipによって参照されるソースパケットは、Clip2の中のどのソースパケットでもよい。それは、Aligned unitの境界である必

要はない。

PlayItemの制限について説明すると、先行するPlayItemのOUT_time (図88、図89において示されるOUT_time1) は、TS1の最後のビデオプレゼンテーションユニットの表示終了時刻を示さなければならない。現在のPlayItemのIN_time (図88、図89において示されるIN_time2) は、TS2の最初のビデオプレゼンテーションユニットの表示開始時刻を示さなければならない。

Bridge-Clipを使用する場合のデータアロケーションの制限について、図94を参照して説明すると、シームレス接続は、ファイルシステムによってデータの連続供給が保証されるように作られなければならない。これは、Clip1 (Clip AVストリームファイル) とClip2 (Clip AVストリームファイル) に接続されるBridge-Clip AVストリームを、データアロケーション規定を満たすように配置することによって行われなければならない。

RSPN_exit_from_previous_Clip以前のClip1 (Clip AVストリームファイル) のストリーム部分が、ハーフフラグメント以上の連続領域に配置されているように、RSPN_exit_from_previous_Clipが選択されなければならない。Bridge-Clip AVストリームのデータ長は、ハーフフラグメント以上の連続領域に配置されるように、選択されなければならない。RSPN_enter_to_current_Clip以後のClip2 (Clip AVストリームファイル) のストリーム部分が、ハーフフラグメント以上の連続領域に配置されているように、RSPN_enter_to_current_Clipが選択されなければならない。

Bridge-Clipを使用しないでシームレス接続する場合のデータアロケーションの制限について、図95を参照して説明すると、シームレス接続は、ファイルシステムによってデータの連続供給が保証されるように作られなければならない。これは、Clip1 (Clip AVストリームファイル) の最後の部分とClip2 (Clip AVストリームファイル) の最初の部分を、データアロケーション規定を満たすように配置することによって行われなければならない。

Clip1 (Clip AVストリームファイル) の最後のストリーム部分が、ハーフフラグメント以上の連続領域に配置されていなければならぬ。Clip2 (Clip AVストリームファイル) の最初のストリーム部分が、ハーフフラグメント以上の連

続領域に配置されていなければならない。

次に、DVR-STDについて説明する。DVR-STDは、DVR MPEG-2トランSPORTストリームの生成及び検証の際ににおけるデコード処理をモデル化するための概念モデルである。また、DVR-STDは、上述したシームレス接続された2つのPlayItemによって参照されるAVストリームの生成及び検証の際ににおけるデコード処理をモデル化するための概念モデルでもある。

DVR-STDモデルを図9.6に示す。図9.6に示したモデルには、DVR MPEG-2トランSPORTストリームプレーヤモデルが構成要素として含まれている。 n , TBn , MBn , EBn , $TBsys$, $Bsys$, Rxn , $Rbxn$, $Rxsys$, Dn , $Dsys$, On 及び $Pn(k)$ の表記方法は、ISO/IEC13818-1のT-STDに定義されているものと同じである。すなわち、次の通りである。 n は、エレメンタリーストリームのインデクス番号である。 TBn は、エレメンタリーストリーム n のトランSPORTバッファである。

MBn は、エレメンタリーストリーム n の多重バッファである。ビデオストリームについてのみ存在する。 EBn は、エレメンタリーストリーム n のエレメンタリーストリームバッファである。ビデオストリームについてのみ存在する。 $TBsys$ は、復号中のプログラムのシステム情報のための入力バッファである。 $Bsys$ は、復号中のプログラムのシステム情報のためのシステムターゲットデコーダ内のメインバッファである。 Rxn は、データが TBn から取り除かれる伝送レートである。 $Rbxn$ は、PESパケットペイロードが MBn から取り除かれる伝送レートである。ビデオストリームについてのみ存在する。

$Rxsys$ は、データが $TBsys$ から取り除かれる伝送レートである。 Dn は、エレメンタリーストリーム n のデコーダである。 $Dsys$ は、復号中のプログラムのシステム情報に関するデコーダである。 On は、ビデオストリーム n のre-ordering bufferである。 $Pn(k)$ は、エレメンタリーストリーム n の k 番目のプレゼンテーションユニットである。

DVR-STDのデコーディングプロセスについて説明する。単一のDVR MPEG-2トランSPORTストリームを再生している間は、トランSPORTパケットを $TB1$, TBn 又は $TBsys$ のバッファへ入力するタイミングは、ソースパケットの

arrival_time_stampにより決定される。TB1, MB1, EB1, TBn, Bn, TBsys及びBsysのバッファリング動作の規定は、ISO/IEC 13818-1に規定されているT-STDと同じである。復号動作と表示動作の規定もまた、ISO/IEC 13818-1に規定されているT-STDと同じである。

シームレス接続されたPlayItemを再生している間のデコーディングプロセスについて説明する。ここでは、シームレス接続されたPlayItemによって参照される2つのAVストリームの再生について説明をすることにし、以後の説明では、上述した（例えば、図88に示した）TS1とTS2の再生について説明する。TS1は、先行するストリームであり、TS2は、現在のストリームである。

図97は、あるAVストリーム（TS1）からそれにシームレスに接続された次のAVストリーム（TS2）へと移るときのトランスポートパケットの入力、復号、表示のタイミングチャートを示す。所定のAVストリーム（TS1）からそれにシームレスに接続された次のAVストリーム（TS2）へと移る間には、TS2のアライバルタイムベースの時間軸（図97においてATC2で示される）は、TS1のアライバルタイムベースの時間軸（図97においてATC1で示される）と同じでない。

また、TS2のシステムタイムベースの時間軸（図97においてSTC2で示される）は、TS1のシステムタイムベースの時間軸（図97においてSTC1で示される）と同じでない。ビデオの表示は、シームレスに連続していることが要求される。オーディオのプレゼンテーションユニットの表示時間にはオーバーラップがあってもよい。

DVR-STDへの入力タイミングについて説明する。時刻T1までの時間、すなわち、TS1の最後のビデオパケットがDVR-STDのTB1に入力終了するまでは、DVR-STDのTB1、TBn又はTBsysのバッファへの入力タイミングは、TS1のソースパケットのarrival_time_stampによって決定される。

TS1の残りのパケットは、TS_recording_rate (TS1) のビットレートでDVR-STDのTBn又はTBsysのバッファへ入力されなければならない。ここで、TS_recording_rate (TS1) は、Clip1に対応するClipInfo () において定義されるTS_recording_rateの値である。TS1の最後のバイトがバッファへ入力する

時刻は、時刻 T 2である。したがって、時刻 T 1から T 2までの区間では、ソースパケットのarrival_time_stampは無視される。

N 1をT S 1の最後のビデオパケットに続くT S 1のトランスポートパケットのバイト数とすると、時刻 T 1乃至 T 2までの時間D T 1は、N 1バイトがTS_recording_rate (T S 1) のビットレートで入力終了するために必要な時間であり、次式により算出される。

$$\Delta T_1 = T_2 - T_1 = N_1 / TS_recording_rate (T S 1)$$

時刻 T 1乃至 T 2までの間は、RXnとRXsysの値は共に、TS_recording_rate (T S 1) の値に変化する。このルール以外のバッファリング動作は、T - STDと同じである。

T 2の時刻において、arrival_time_clock_counterは、T S 2の最初のソースパケットのarrival_time_stampの値にリセットされる。DVR - STDのTB1, TBn 又はTBsysのバッファへの入力タイミングは、T S 2のソースパケットのarrival_time_stampによって決定される。RXnとRXsysは共に、T - STDにおいて定義されている値に変化する。

付加的なオーディオバッファリング及びシステムデータバッファリングについて説明すると、オーディオデコーダとシステムデコーダは、時刻 T 1からT2までの区間の入力データを処理することができるよう、T - STDで定義されるバッファ量に加えて付加的なバッファ量（約 1秒分のデータ量）が必要である。

ビデオのプレゼンテーションタイミングについて説明すると、ビデオプレゼンテーションユニットの表示は、接続点を通して、ギャップなしに連続でなければならない。ここで、STC 1は、T S 1のシステムタイムベースの時間軸（図9 7ではSTC 1と図示されている）とし、STC 2は、T S 2のシステムタイムベースの時間軸（図9 7ではSTC 2と図示されている。正確には、STC 2は、T S 2の最初のPCRがT - STDに入力した時刻から開始する。）とする。

STC 1とSTC 2の間のオフセットは、次のように決定される。PTS1endは、T S 1の最後のビデオプレゼンテーションユニットに対応するSTC 1上のPTSであり、PTS2startは、T S 2の最初のビデオプレゼンテーションユニットに対応するSTC 2上のPTSであり、Tppは、T S 1の最後のビデオプレゼンテー

ョンユニットの表示期間とすると、2つのシステムタイムベースの間のオフセット STC_deltaは、次式により算出される。

$$\text{STC_delta} = \text{PTS1end} + \text{Tpp} - \text{PTS2start}$$

オーディオのプレゼンテーションのタイミングについて説明すると、接続点において、オーディオプレゼンテーションユニットの表示タイミングのオーバーラップがあつても良く、それは0乃至2オーディオフレーム未満である（図97に図示されている”audio overlap”を参照）。どちらのオーディオサンプルを選択するかということと、オーディオプレゼンテーションユニットの表示を接続点の後の補正されたタイムベースに再同期することは、プレーヤ側により設定されることである。

DVR-STDのシステムタイムクロックについて説明すると、時刻T5において、TS1の最後のオーディオプレゼンテーションユニットが表示される。システムタイムクロックは、時刻T2からT5の間にオーバーラップしていてもよい。この区間では、DVR-STDは、システムタイムクロックを古いタイムベースの値(STC1)と新しいタイムベースの値(STC2)の間で切り替える。STC2の値は、次式により算出される。

$$\text{STC2} = \text{STC1} - \text{STC_delta}$$

バッファリングの連続性について説明する。STC11video_endは、TS1の最後のビデオパケットの最後のバイトがDVR-STDのTB1へ到着するときのシステムタイムベースSTC1上のSTCの値である。STC22video_startは、TS2の最初のビデオパケットの最初のバイトがDVR-STDのTB1へ到着するときのシステムタイムベースSTC2上のSTCの値である。STC21video_endは、STC11video_endの値をシステムタイムベースSTC2上の値に換算した値である。STC21video_endは、次式により算出される。

$$\text{STC21video_end} = \text{STC11video_end} - \text{STC_delta}$$

DVR-STDに従うために、次の2つの条件を満たすことが要求される。まず、TS2の最初のビデオパケットのTB1への到着タイミングは、次に示す不等式を満たさなければならない。そして、次に示す不等式を満たさなければならない。

STC22video_start > STC21video_end + ΔT1

この不等式が満たされるように、Clip1及び、又は、Clip2の部分的なストリームを再エンコード及び、又は、再多重化する必要がある場合は、その必要に応じて行われる。

次に、STC1とSTC2を同じ時間軸上に換算したシステムタイムベースの時間軸上において、TS1からのビデオパケットの入力とそれに続くTS2からのビデオパケットの入力は、ビデオバッファをオーバーフロウ及びアンダーフローさせてはならない。

このようなシンタクス、データ構造、規則に基づくことにより、記録媒体に記録されているデータの内容、再生情報等を適切に管理することができ、もって、ユーザが再生時に適切に記録媒体に記録されているデータの内容を確認したり、所望のデータを簡便に再生できるようにすることができる。

なお、本例は、多重化ストリームとしてMPEG2トランスポートストリームを例にして説明しているが、これに限らず、MPEG2プログラムストリームや米国のDirectTVサービス（商標）で使用されているDSSトランスポートストリームについても適用することが可能である。

図98は、Info.dvrの作成又は更新の処理を説明するフローチャートを示す。

図1の記録再生装置1のブロック図を参照しながら説明する。

ステップS11で、ユーザがディスクに再生制限をかける場合、制御部23はユーザインタフェースを通して、PINを取得する。

ステップS12で、制御部23は、ユーザインタフェースを通して、ディスク名又は/及びディスクを代表するサムネイルを取得する。

ステップS13で、制御部23は、新規に記録されたPlayListのファイル名をTableOfPlayListへストアする。

ステップS14で、制御部23は、最後に再生したPlayListのファイル名を取得する。これは、resume_PlayList_nameへストアされる。

ステップS15で、制御部23は、メーカの特別なアプリケーションのためのメーカプライベートデータを取得する。これは、MakersPrivateDataへストアされる。

ステップS16で、制御部23は、最後に再生したPlayListの再生中断時刻を取得する。これはPlayListファイルのPlayListMarkのリジュームマーカーへストアされる。

ステップS17で、制御部23は、ユーザインタフェースを通して、ユーザにPlayListの再生制限をかけるかを確認する。制御部23は、ユーザが再生制限をかけることを指示したPlayListのUIAppInfoPlayListのplayback_control_flagをセットする。

ステップS18で、制御部23は、info.dvrをディスクに記録するように指示する。

ステップS19で、制御部23は、変更された、又は新規に作られたPlayListファイルをディスクに記録するように指示する。

図99は、ディスクの記録内容をユーザインタフェースへ提示する処理を説明するフローチャートを示す。図1の記録再生装置1のブロック図を参照しながら説明する。

ステップS31で、制御部23はディスクに記録されているinfo.dvr, Clip Information file, PlayList file及びThumbnail fileの情報を取得する。

ステップS32で、制御部23はVolume_protect_flagがセットされている場合は、ユーザインタフェースを通して、PINの入力をユーザに要求する。

ステップS33で、制御部23はディスク名及びディスク内容を代表するサムネイルをユーザインタフェースを通して、GUIに提示する。

ステップS34で、制御部23は、TableOfPlayListにエントリされている順番にPlayListの名前を並べたPlayList一覧画面を、GUIに提示する。

ステップS35で、制御部23は、リジューム(Resume)できるPlayListがある場合は、それをGUIに提示する。

ステップS36で、ユーザインタフェースを通して、ユーザが1つのPlayListの再生を指示する。

ステップS37で、制御部23は、指示されたPlayListのplayback_control_flagがセットされている場合は、ユーザインタフェースを通して、PINの入力をユーザに要求する。

ステップS38で、制御部23は、PlayListの再生開始点を先頭時刻又はresumeマーカーの時刻のどちらにするかをユーザに確認する。

ステップS39で、制御部23は、指示された時刻からPlayListを再生する。

このようにして、ディスクの記録内容をユーザインタフェースへ提示し、ユーザは再生したい1つのPlayListを選択し、プレーヤは選択されたPlayListを再生する。

図100は、TableOfPlayListに複数のPlayListがエントリされている場合、それらPlayListの再生順序を変更する処理を説明するフローチャートである。

図1の記録再生装置1のブロック図を参照しながら説明する。

ステップS51で、制御部23はディスクに記録されているinfo.dvr, Clip Information file, PlayList file及びThumbnail fileの情報を取得する。

ステップS52で、制御部23は、TableOfPlayListにエントリされている順番にPlayListの名前を並べたPlayList一覧画面を、GUIに提示する。

ステップS53で、ユーザがユーザインタフェースを通して、PlayListの再生順序の変更を指示する。

ステップS54で、制御部23は、上記の新しい順序に基づいてTableOfPlayListを更新する。

ステップS55で、制御部23は、info.dvrをディスクに記録することを指示する。

このようにして、ディスクに複数のPlayListが記録されている場合に、それらのデフォルトの再生順序をTableOfPlayListに記録することができるようにより、ユーザはこの再生順序を自由に設定できる。

このようなシンタクス、データ構造、規則に基づくことにより、記録媒体に記録されているデータの内容、再生情報等を適切に管理することができ、もって、ユーザが再生時に適切に記録媒体に記録されているデータの内容を確認したり、所望のデータを簡便に再生できるようにすることができる。すなわち、Info.dvrを用いて、ユーザが所望のPlayListを選択する際の処理を簡便化することができる。

また、Info.dvrのファイルをPlayListファイルやClip Informationファイルと

は分離して記録することにより、Info.dvrのファイルサイズを非常に小さくできる。そのため、Info.dvrファイルの内容を変更して、それを記録するときに必要な時間を小さくできる。また、Info.dvrの変更に関係のないPlayListファイルやClip Informationファイルを変更する必要がない。

もし、Info.dvr、PlayList及びClip Informationの内容を1つのファイルにして記録すると、ファイルサイズは非常に大きくなる。そのために、そのファイルの内容を変更して、それを記録するためにかかる時間は、Info.dvrだけを1つのファイルで記録する場合に比べて、非常に大きくなる。

新規にPlayListを記録する場合（図98）やユーザがPlayListの再生順序を変更するという場合（図1.00）等において、Info.dvrファイルが書きかえられる。Info.dvrの書き換えは、PlayListファイルやClip Informationファイルに比べて、頻繁に行われる所以、Info.dvrを1つのファイルにして管理することは、この書き換え時の処理時間の短縮に効果が大きい。

また、Info.dvrは小さなファイルであるので、ディスクから読み出す時間も小さい。最初にInfo.dvrだけを読み出して、その内容に基づいて、ディスクの記録内容をユーザインタフェースへ提示する場合（図99）、ユーザの待ち時間を小さくすることができる。

上述した一連の処理は、ハードウェアにより実行させることもできるが、ソフトウェアにより実行させることもできる。一連の処理をソフトウェアにより実行せる場合には、そのソフトウェアを構成するプログラムが専用のハードウェアに組み込まれているコンピュータ、又は、各種のプログラムをインストールすることで、各種の機能を実行することが可能な、例えば汎用のパーソナルコンピュータ等に、記録媒体からインストールされる。

この記録媒体は、図1.01に示すように、コンピュータとは別に、ユーザにプログラムを提供するために配布される、プログラムが記録されている磁気ディスク221（フロッピディスクを含む）、光ディスク222（CD-ROM（Compact Disk-Read Only Memory）、DVD（Digital Versatile Disk）を含む）、光磁気ディスク223（MD（Mini-Disk）を含む）、若しくは半導体メモリ224等よりなるパッケージメディアにより構成されるだけでなく、コンピュータに予

め組み込まれた状態でユーザーに提供される、プログラムが記憶されているROM 202や記憶部208が含まれるハードディスク等で構成される。

なお、本明細書において、媒体により提供されるプログラムを記述するステップは、記載された順序に従って、時系列的に行われる処理は勿論、必ずしも時系列的に処理されなくとも、並列的或いは個別に実行される処理をも含むものである。

また、本明細書において、システムとは、複数の装置により構成される装置全体を表すものである。

産業上の利用可能性

以上のごとく、本発明に係る情報処理装置及び方法、記録媒体、並びにプログラム、並びに第2の情報処理装置及び方法、記録媒体、並びにプログラムによれば、管理情報が、再生指定情報に基づく再生が終了された時点で再生指定情報に付けられていた名称に関する名称情報を含み、再生指定情報が、再生指定情報に基づく再生が終了された時点の時間情報を含むようにすることができる。

本発明に係る情報処理装置及び方法、記録媒体、並びにプログラム、並びに第4の情報処理装置及び方法、記録媒体、並びにプログラムによれば、管理情報が、管理情報が管理する全ての再生指定情報についての閲覧の許可に関する閲覧許可情報を含み、再生指定情報が、再生指定情報の閲覧の許可に関する閲覧許可情報を含むようすることができる。

本発明に係る情報処理装置及び方法、記録媒体、並びにプログラムによれば、管理情報が、管理情報が管理する全ての再生指定情報を再生順に登録する再生順序情報を含み、再生指定情報が、その再生区間の時間情報を含むようにすることができる。

したがって、いずれの場合においても、記録媒体に記録されているデータ内容、及び、再生情報を適切に管理することができる。

請求の範囲

1. 記録媒体に記録されている情報の再生手順を指定する再生指定情報と、前記再生指定情報を管理する管理情報を生成する生成手段と、
前記生成手段により生成された前記再生指定情報と前記管理情報を前記記録媒体に記録する記録手段とを有し、
前記管理情報は、前記再生指定情報に基づく再生が終了された時点で前記再生指定情報に付けられていた名称に関する名称情報を含み、
前記再生指定情報は、前記再生指定情報に基づく再生が終了された時点の時間情報を含む情報処理装置。
2. 記録媒体に記録されている情報の再生手順を指定する再生指定情報と、前記再生指定情報を管理する管理情報を生成する生成ステップと、
前記生成ステップの処理により生成された前記再生指定情報と前記管理情報を前記記録媒体に記録する記録ステップとを含み、
前記管理情報は、前記再生指定情報に基づく再生が終了された時点で前記再生指定情報に付けられていた名称に関する名称情報を含み、
前記再生指定情報は、前記再生指定情報に基づく再生が終了された時点の時間情報を含む情報処理方法。
3. 記録媒体に記録されている情報の再生手順を指定する再生指定情報と、前記再生指定情報を管理する管理情報を生成する生成ステップと、
前記生成ステップの処理により生成された前記再生指定情報と前記管理情報を前記記録媒体に記録する記録ステップとを含むプログラムにおいて、
前記管理情報は、前記再生指定情報に基づく再生が終了された時点で前記再生指定情報に付けられていた名称に関する名称情報を含み、
前記再生指定情報は、前記再生指定情報に基づく再生が終了された時点の時間情報を含むコンピュータが読み取り可能なプログラムが記録されている記録媒体。
4. 記録媒体に記録されている情報の再生手順を指定する再生指定情報と、前記再生指定情報を管理する管理情報を生成する生成ステップと、

前記生成ステップの処理により生成された前記再生指定情報と前記管理情報を前記記録媒体に記録する記録ステップとをコンピュータに実行させるプログラムにおいて、

前記管理情報は、前記再生指定情報に基づく再生が終了された時点で前記再生指定情報に付けられていた名称に関する名称情報を含み、

前記再生指定情報は、前記再生指定情報に基づく再生が終了された時点の時間情報を含むプログラム。

5. 記録されている主情報の再生手順を指定する再生指定情報と、前記再生指定情報を管理する管理情報が記録されている記録媒体から、前記主情報を再生する情報処理装置において、

前記再生指定情報に基づく再生が終了された時点で前記再生指定情報に付けられていた名称に関する名称情報を含む前記管理情報と、前記再生指定情報に基づく再生が終了された時点の時間情報を含む前記再生指定情報に基づいて、前記記録媒体の主情報の再生を制御する制御手段を備える情報処理装置。

6. 記録されている主情報の再生手順を指定する再生指定情報と、前記再生指定情報を管理する管理情報が記録されている記録媒体から、前記主情報を再生する情報処理装置の情報再生方法において、

前記再生指定情報に基づく再生が終了された時点で前記再生指定情報に付けられていた名称に関する名称情報を含む前記管理情報と、前記再生指定情報に基づく再生が終了された時点の時間情報を含む前記再生指定情報に基づいて、前記記録媒体の主情報の再生を制御する制御ステップを含む情報処理方法。

7. 記録されている主情報の再生手順を指定する再生指定情報と、前記再生指定情報を管理する管理情報が記録されている記録媒体から、前記主情報を再生する情報処理装置のプログラムにおいて、

前記再生指定情報に基づく再生が終了された時点で前記再生指定情報に付けられていた名称に関する名称情報を含む前記管理情報と、前記再生指定情報に基づく再生が終了された時点の時間情報を含む前記再生指定情報に基づいて、前記記録媒体の主情報の再生を制御する制御ステップを含むコンピュータが読み取り可能なプログラムが記録されている記録媒体。

8. 記録されている主情報の再生手順を指定する再生指定情報と、前記再生指定情報を管理する管理情報が記録されている記録媒体から、前記主情報を再生する情報処理装置を制御するコンピュータに、

前記再生指定情報に基づく再生が終了された時点で前記再生指定情報に付けられていた名称に関する名称情報を含む前記管理情報と、前記再生指定情報に基づく再生が終了された時点の時間情報を含む前記再生指定情報に基づいて、前記記録媒体の主情報を再生を制御する制御ステップを実行させるプログラム。

9. 主情報と、記録されている主情報の再生手順を指定する再生指定情報と、前記再生指定情報を管理する管理情報が記録されている記録媒体において、

前記管理情報は、前記再生指定情報に基づく再生が終了された時点で前記再生指定情報に付けられていた名称に関する名称情報を含み、

前記再生指定情報は、前記再生指定情報に基づく再生が終了された時点の時間情報を含む記録媒体。

10. 記録媒体に記録されている情報の再生手順を指定する再生指定情報と、前記再生指定情報を管理する管理情報を生成する生成手段と、

前記生成手段により生成された前記再生指定情報と前記管理情報を前記記録媒体に記録する記録手段とを有し、

前記管理情報は、前記管理情報が管理する全ての再生指定情報についての閲覧の許可に関する閲覧許可情報を含み、

前記再生指定情報は、前記再生指定情報の閲覧の許可に関する閲覧許可情報を含む情報処理装置。

11. 記録媒体に記録されている情報の再生手順を指定する再生指定情報と、前記再生指定情報を管理する管理情報を生成する生成ステップと、

前記生成ステップの処理により生成された前記再生指定情報と前記管理情報を前記記録媒体に記録する記録ステップとを含み、

前記管理情報は、前記管理情報が管理する全ての再生指定情報についての閲覧の許可に関する閲覧許可情報を含み、

前記再生指定情報は、前記再生指定情報の閲覧の許可に関する閲覧許可情報を含む情報処理方法。

12. 記録媒体に記録されている情報の再生手順を指定する再生指定情報と、前記再生指定情報を管理する管理情報を生成する生成ステップと、

前記生成ステップの処理により生成された前記再生指定情報と前記管理情報を前記記録媒体に記録する記録ステップとを含むプログラムにおいて、

前記管理情報は、前記管理情報が管理する全ての再生指定情報についての閲覧の許可に関する閲覧許可情報を含み、

前記再生指定情報は、前記再生指定情報の閲覧の許可に関する閲覧許可情報を含むコンピュータが読み取り可能なプログラムが記録されている記録媒体。

13. 記録媒体に記録されている情報の再生手順を指定する再生指定情報と、前記再生指定情報を管理する管理情報を生成する生成ステップと、

前記生成ステップの処理により生成された前記再生指定情報と前記管理情報を前記記録媒体に記録する記録ステップとをコンピュータに実行させるプログラムにおいて、

前記管理情報は、前記管理情報が管理する全ての再生指定情報についての閲覧の許可に関する閲覧許可情報を含み、

前記再生指定情報は、前記再生指定情報の閲覧の許可に関する閲覧許可情報を含むプログラム。

14. 主情報と、記録されている主情報の再生手順を指定する再生指定情報と、前記再生指定情報を管理する管理情報が記録されている記録媒体において、

前記管理情報は、前記管理情報が管理する全ての再生指定情報についての閲覧の許可に関する閲覧許可情報を含み、

前記再生指定情報は、前記再生指定情報の閲覧の許可に関する閲覧許可情報を含む記録媒体。

15. 記録媒体に記録されている情報の再生手順を指定する再生指定情報と、前記再生指定情報を管理する管理情報を生成する生成手段と、

前記生成手段により生成された前記管理情報を前記記録媒体に記録する記録手段とを有し、

前記管理情報は、前記管理情報が管理する全ての前記再生指定情報を再生順に登録する再生順序情報を含み、

前記再生指定情報は、その再生区間の時間情報を含む情報処理装置。

16. 記録媒体に記録されている情報の再生手順を指定する再生指定情報と、前記再生指定情報を管理する管理情報を生成する生成ステップと、

前記生成ステップの処理により生成された前記管理情報を前記記録媒体に記録する記録ステップとを含み、

前記管理情報は、前記管理情報が管理する全ての前記再生指定情報を再生順に登録する再生順序情報を含み、

前記再生指定情報は、その再生区間の時間情報を含む情報処理方法。

17. 記録媒体に記録されている情報の再生手順を指定する再生指定情報と、前記再生指定情報を管理する管理情報を生成する生成ステップと、

前記生成ステップの処理により生成された前記管理情報を前記記録媒体に記録する記録ステップとを含むプログラムにおいて、

前記管理情報は、前記管理情報が管理する全ての前記再生指定情報を再生順に登録する再生順序情報を含み、

前記再生指定情報は、その再生区間の時間情報を含むコンピュータが読み取り可能なプログラムが記録されている記録媒体。

18. 記録媒体に記録されている情報の再生手順を指定する再生指定情報と、前記再生指定情報を管理する管理情報を生成する生成ステップと、

前記生成ステップの処理により生成された前記管理情報を前記記録媒体に記録する記録ステップとをコンピュータに実行させるプログラムにおいて、

前記管理情報は、前記管理情報が管理する全ての前記再生指定情報を再生順に登録する再生順序情報を含み、

前記再生指定情報は、その再生区間の時間情報を含むプログラム。

19. 記録されている主情報の再生手順を指定する再生指定情報と、前記再生指定情報を管理する管理情報が記録されている記録媒体から、前記主情報を再生する情報処理装置において、

前記管理情報が管理する全ての前記再生指定情報を再生順に登録する再生順序情報を含む前記管理情報と、再生区間の時間情報を含む前記再生指定情報に基づいて、前記記録媒体の主情報の再生を制御する制御手段を備える情報処理装置。

20. 記録されている主情報の再生手順を指定する再生指定情報と、前記再生指定情報を管理する管理情報が記録されている記録媒体から、前記主情報を再生する情報処理装置の情報処理方法において、

前記管理情報が管理する全ての前記再生指定情報を再生順に登録する再生順序情報を含む前記管理情報と、再生区間の時間情報を含む前記再生指定情報に基づいて、前記記録媒体の主情報を制御する制御ステップを含む情報処理方法。

21. 記録されている主情報の再生手順を指定する再生指定情報と、前記再生指定情報を管理する管理情報が記録されている記録媒体から、前記主情報を再生する情報処理装置のプログラムにおいて、

前記管理情報が管理する全ての前記再生指定情報を再生順に登録する再生順序情報を含む前記管理情報と、再生区間の時間情報を含む前記再生指定情報に基づいて、前記記録媒体の主情報を制御する制御ステップを含むコンピュータが読み取り可能なプログラムが記録されている記録媒体。

22. 記録されている主情報の再生手順を指定する再生指定情報と、前記再生指定情報を管理する管理情報が記録されている記録媒体から、前記主情報を再生する情報処理装置を制御するコンピュータに、

前記管理情報が管理する全ての前記再生指定情報を再生順に登録する再生順序情報を含む前記管理情報と、再生区間の時間情報を含む前記再生指定情報に基づいて、前記記録媒体の主情報を制御する制御ステップを実行させるプログラム。

23. 主情報と、記録されている主情報の再生手順を指定する再生指定情報と、前記再生指定情報を管理する管理情報が記録されている記録媒体において、

前記管理情報は、前記管理情報が管理する全ての前記再生指定情報を再生順に登録する再生順序情報を含み、

前記再生指定情報は、その再生区間の時間情報を含む記録媒体。

1/93

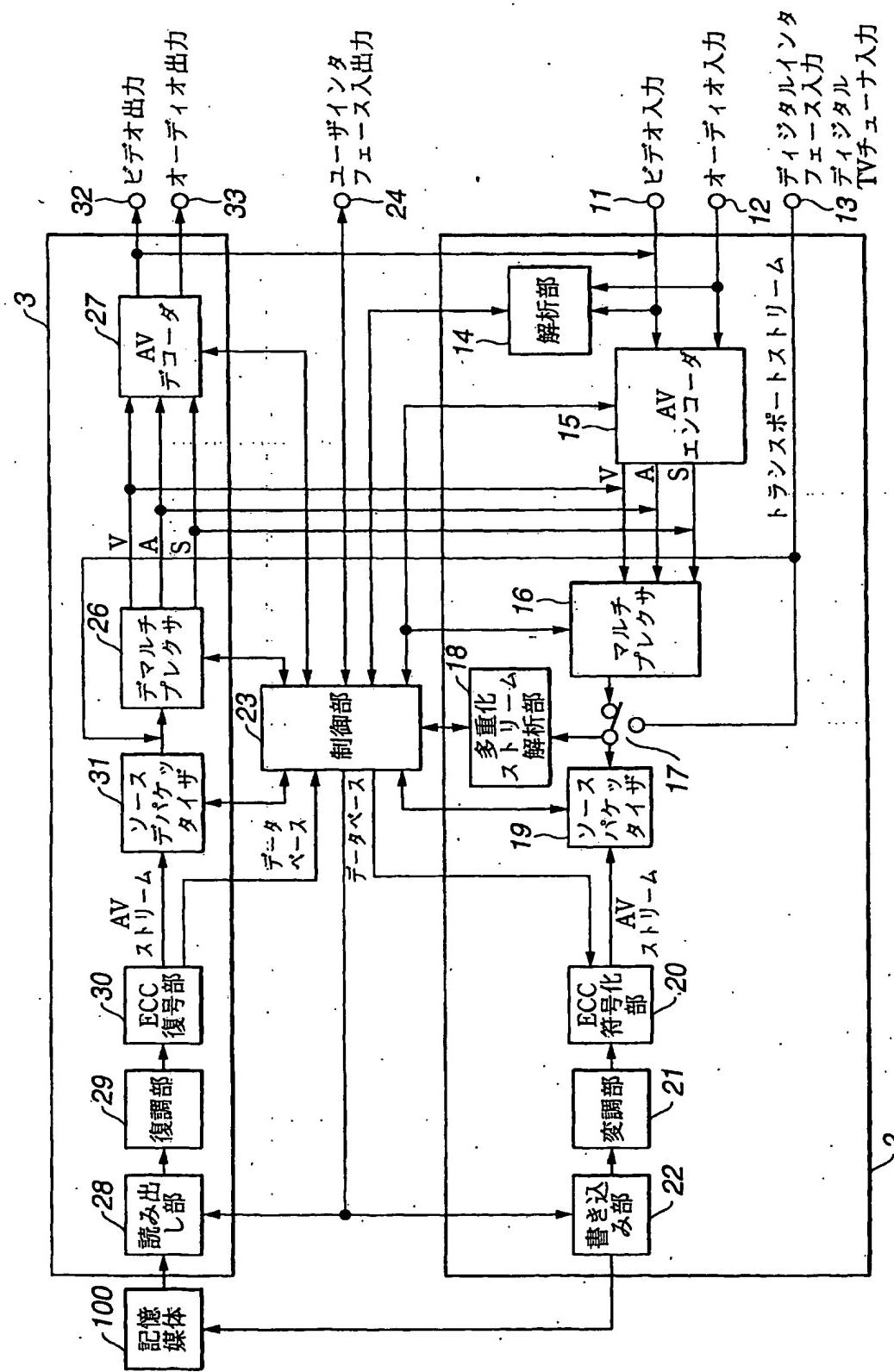
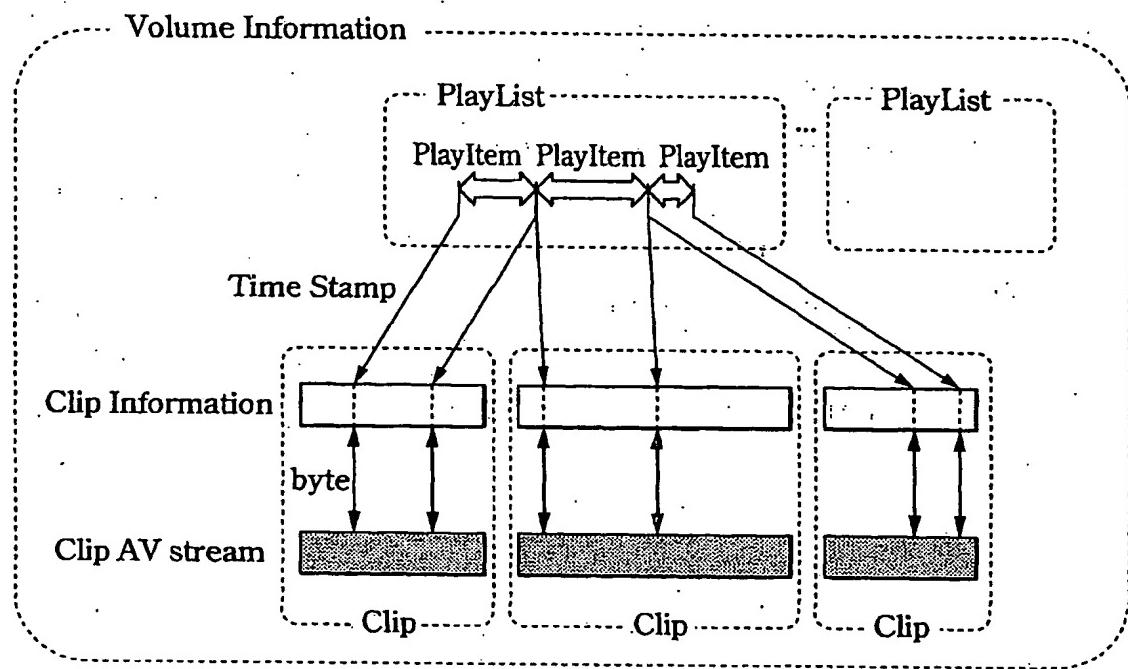


FIG.1

2/93

**FIG.2**

3/93

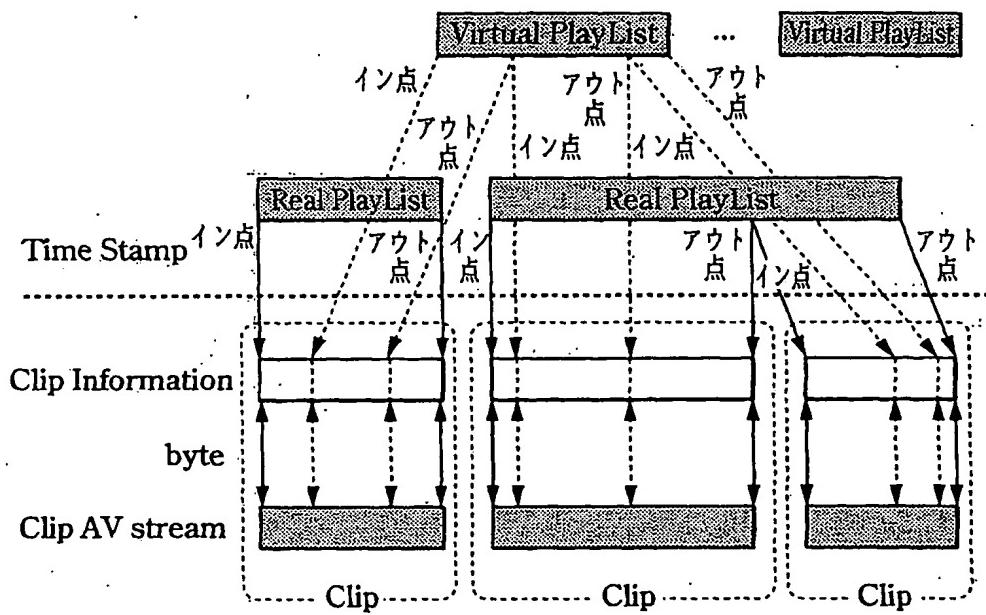
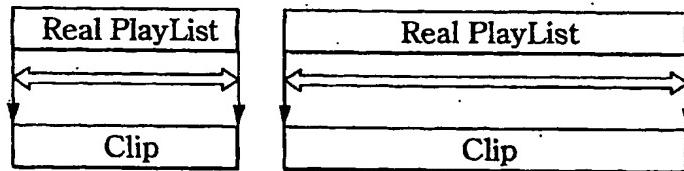
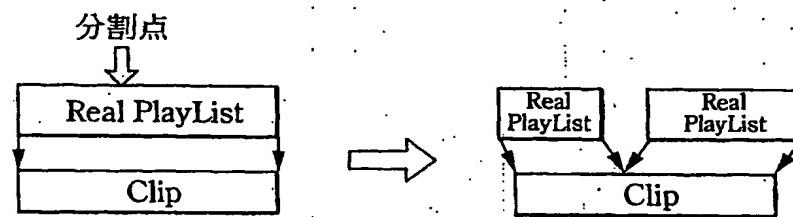
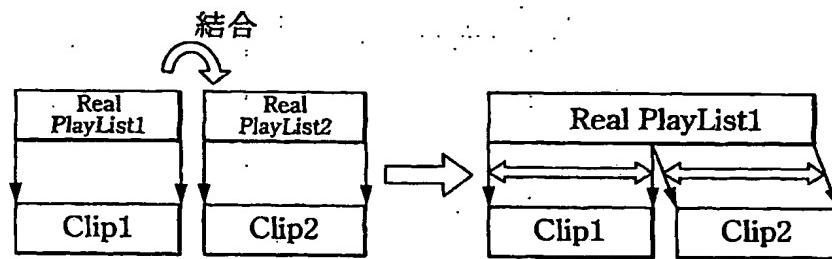
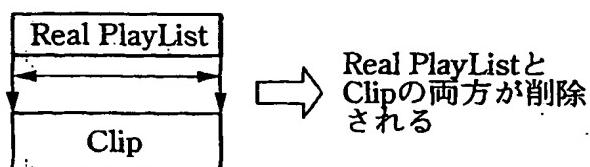
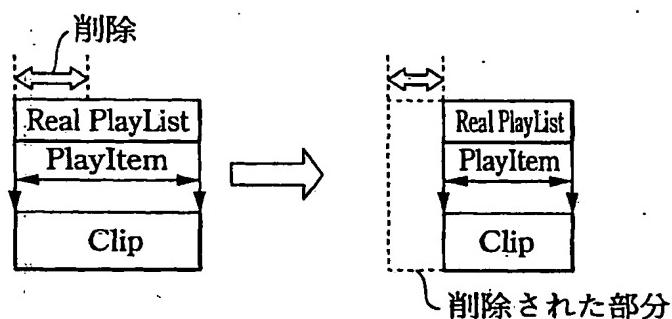
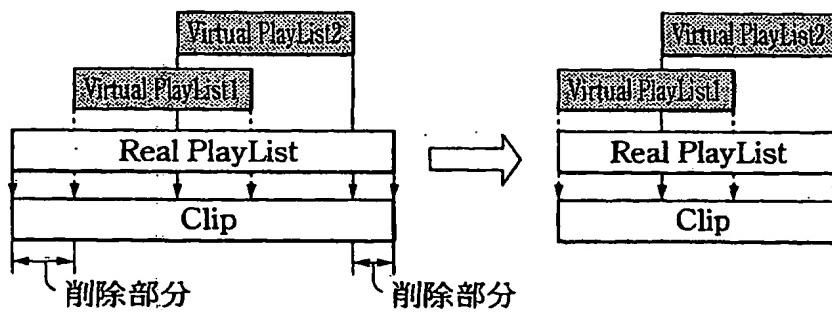


FIG.3

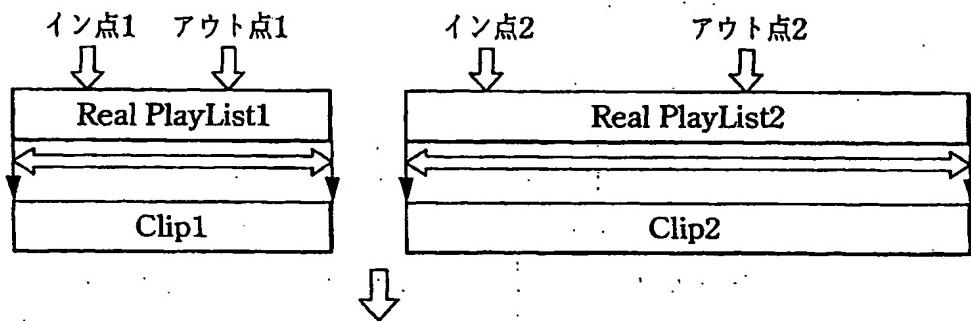
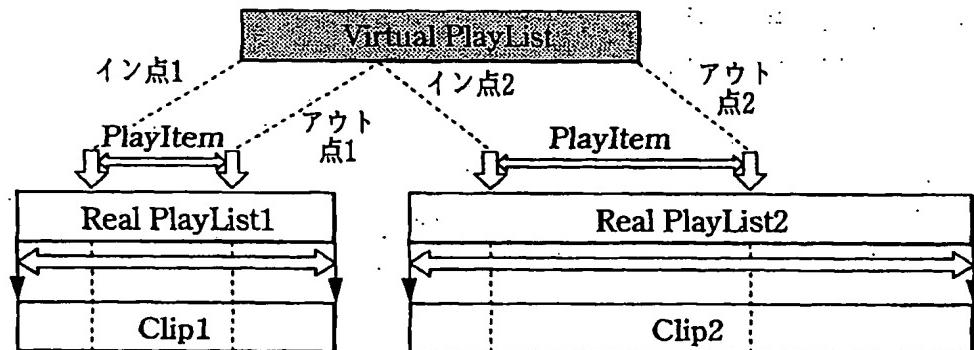
4/93

**FIG.4A****FIG.4B****FIG.4C**

5/93

**FIG.5A****FIG.5B****FIG.5C**

6/93

FIG.6A**FIG.6B**

7/93

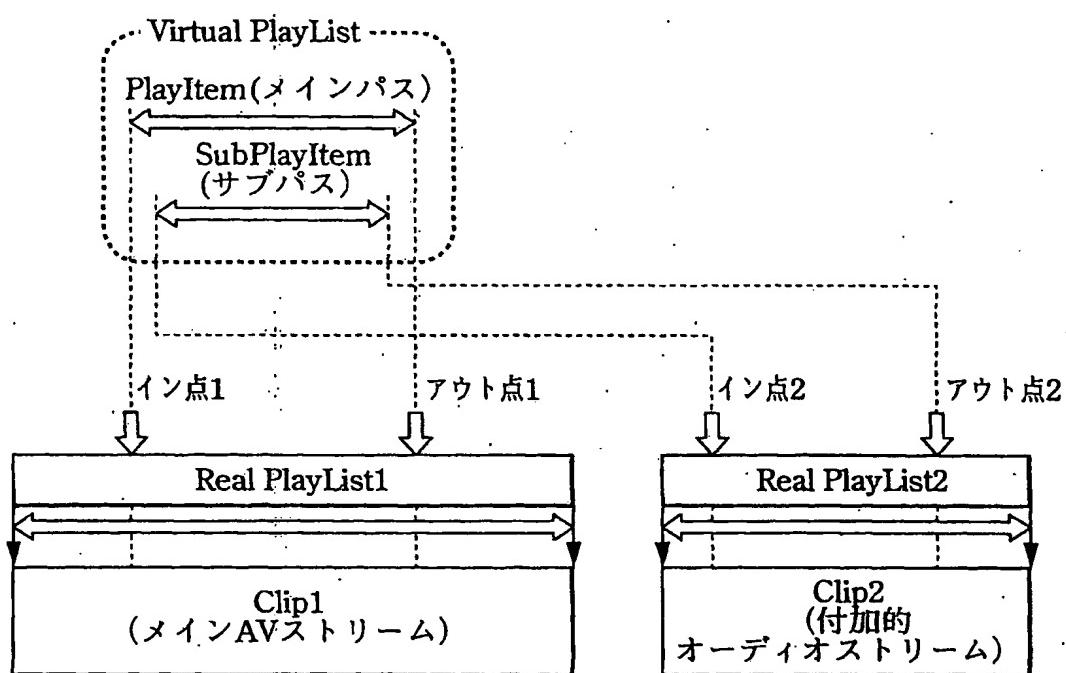
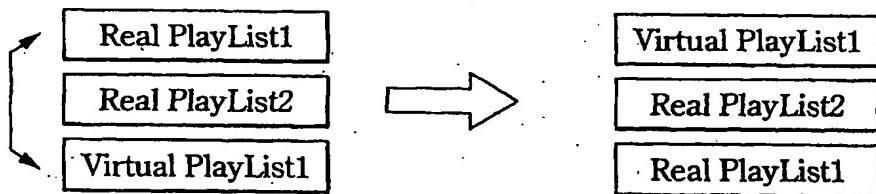


FIG.7

8/93

再生順序

**FIG.8**

9/93

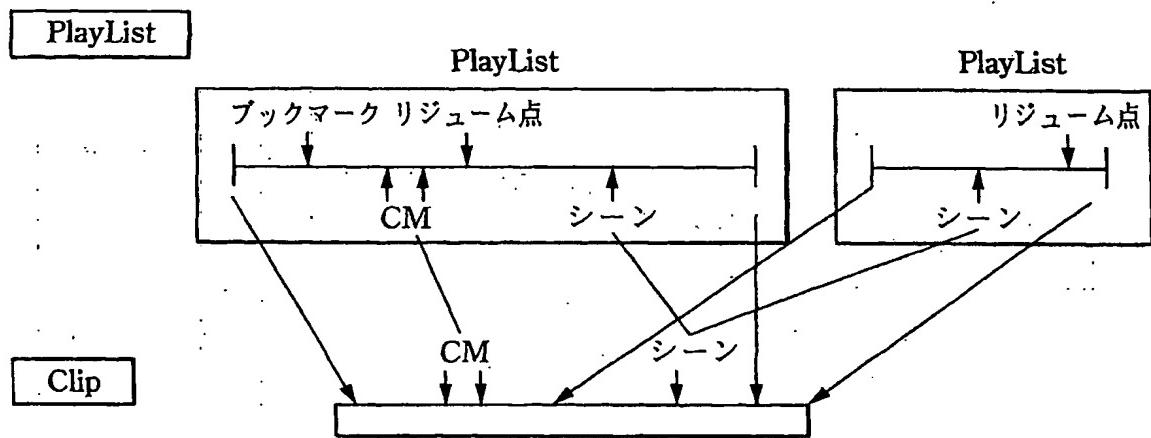


FIG.9

10/93

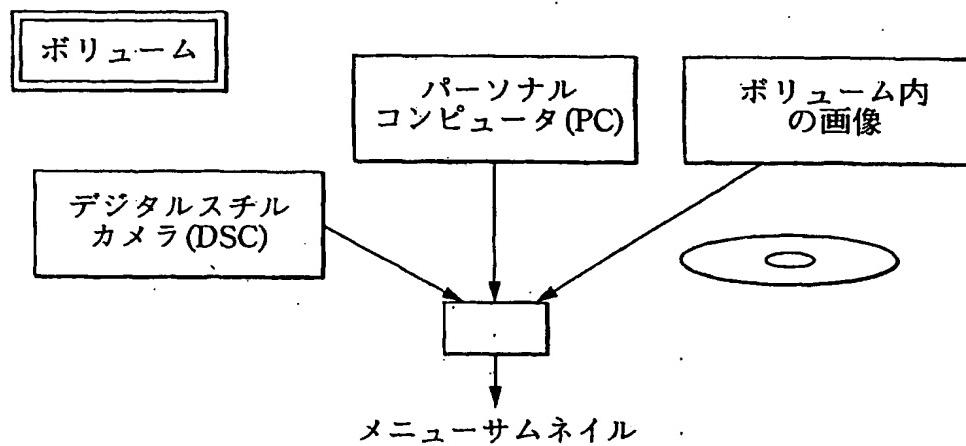


FIG.10

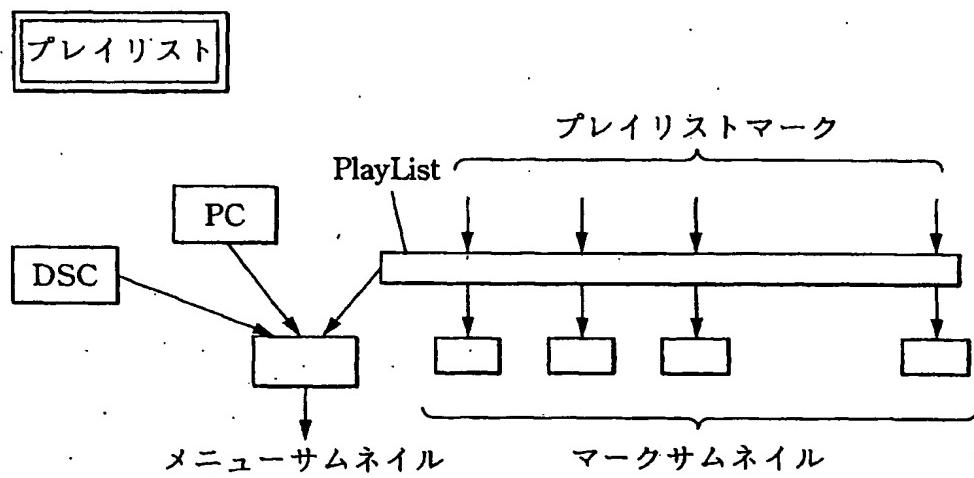


FIG.11

11/93

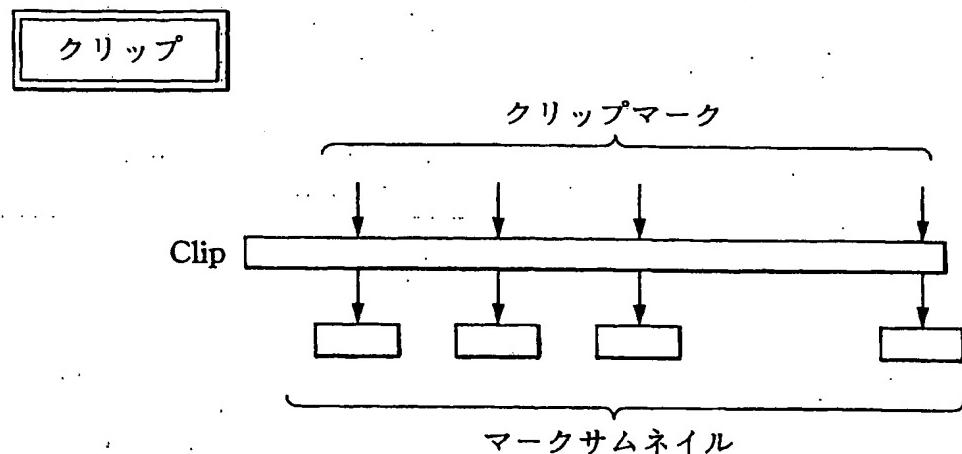


FIG.12

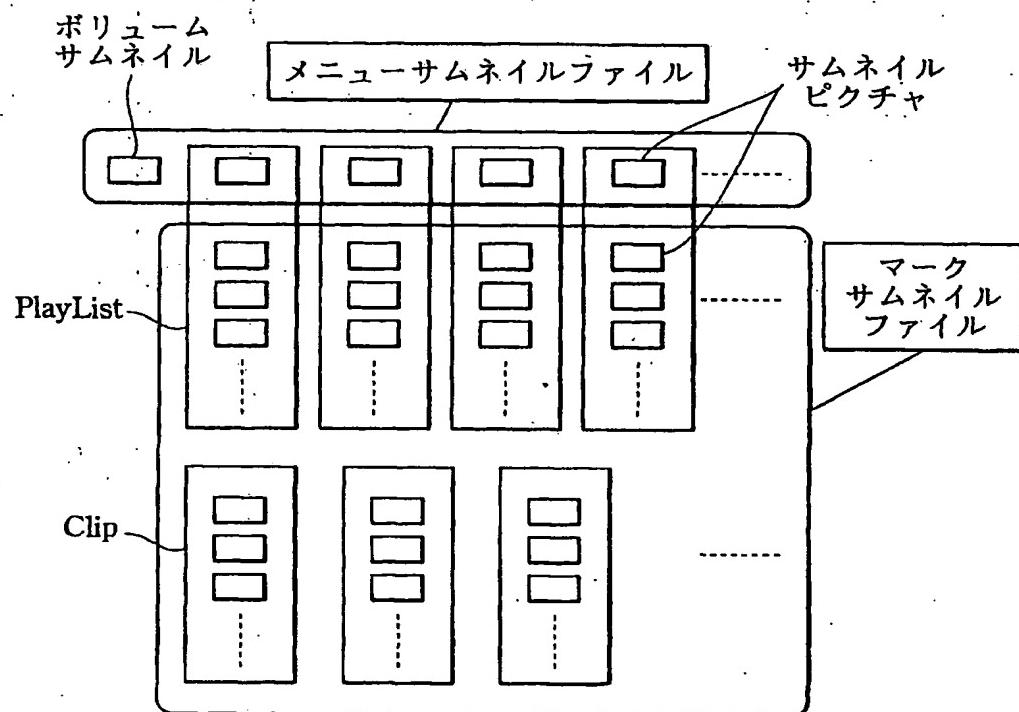
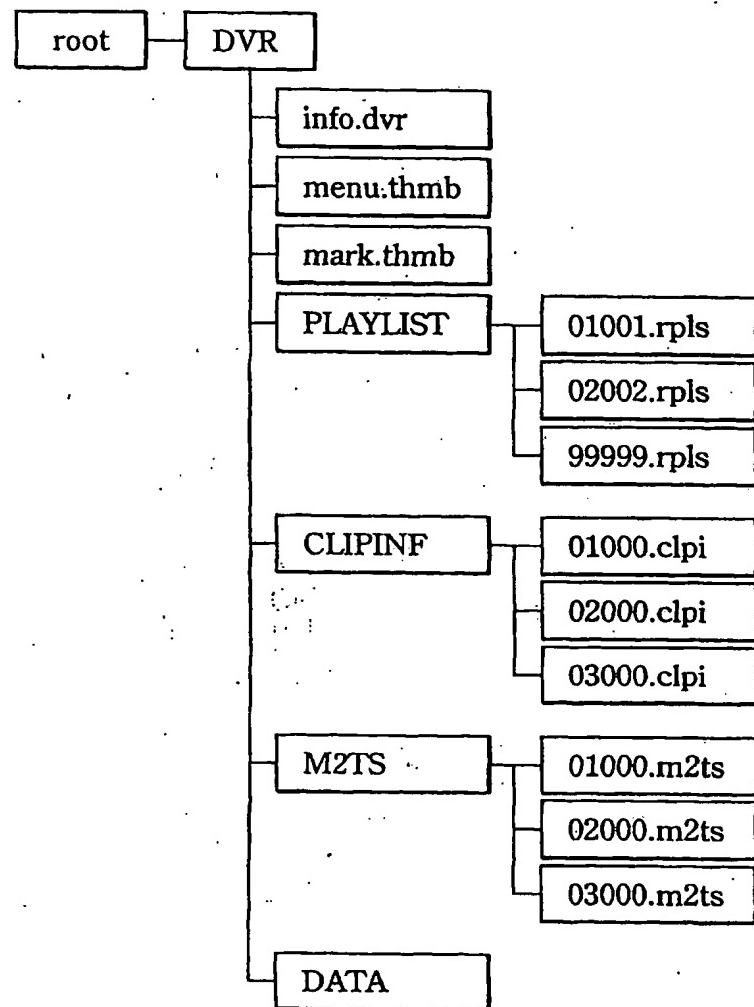


FIG.13

12/93**FIG.14**

13/93

シンタクス	バイト数	略号
info.dvr {		
<u>TableOfPlayLists_Start_address</u>	32	uimsbf
<u>MakersPrivateData_Start_address</u>	32	uimsbf
reserved	192	bslbf
<u>DVRVolume()</u>		
for (i=0;i<N1;i++){		
<u>padding_word</u>	16	bslbf
}		
<u>TableOfPlayLists()</u>		
for (i=0;i<N2;i++){		
<u>padding_word</u>	16	bslbf
}		
<u>MakersPrivateData()</u>		
}		

FIG.15

14/93

シンタクス	バイト数	略号
DVRVolume()		
version_number	8*4	bslbf
length	32	uimsbf
ResumeVolume()		
UIAppInfoVolume()		
}		

FIG.16

15/93

シンタクス	バイト数	略号
ResumeVolume{		
reserved	15	bslbf
valid_flag	1	bslbf
resume_PlayList_name	8*10	bslbf
}		

FIG.17

16/93

シンタクス	バイト数	略号
UIAppInfoVolume()		
character_set	8	bslbf
name_length	8	uimsbf
Volume_name	8*256	bslbf
reserved	15	bslbf
Volume_protect_flag	1	bslbf
PIN	8*4	bslbf
ref_thumbnail_index	16	uimsbf
reserved	7	bslbf
rp_info_valid_flag	1	uimsbf
rp_ref_to_PlayList_file_name	8*10	bslbf
rp_ref_to_PlayItem_id	16	uimsbf
rp_time_stamp	32	uimsbf
}		

FIG.18

17/93

値	キャラクタ文字符串化
0x00	Reserved
0x01	ISO/IEC 646 (ASCII)
0x02	ISO/IEC 10646-1 (Unicode)
0x03-0xff	Reserved

FIG.19

18/93

シンタクス	バイト数	略号
TableOfPlayLists()		
version_number	8*4	bslbf
length	32	uimsbf
number_of_PlayLists	16	uimsbf
for (i=0; i<number_of_PlayLists; i++) {		
PlayList_file_name	8*10	bslbf
}		
}		

FIG.20

19/93

シンタクス	バイト数	略号
TableOfPlayLists()		
version_number	8*4	bslbf
length	32	uimsbf
number_of_PlayLists	16	uimsbf
for (i=0; i<number_of_PlayLists; i++) {		
PlayList_file_name	8*10	bslbf
UIAppInfoPlayList()		
}		
}		

FIG.21

20/93

シンタクス	バイト数	略号
MakersPrivateData()		
version_number	8*4	bslbf
length	32	uimsbf
if (length !=0){		
mpd_blocks_start_address	32	uimsbf
number_of_maker_entries	16	uimsbf
mpd_block_size	16	uimsbf
number_of_mpd_blocks	16	uimsbf
reserved	16	bslbf
for (i=0; i<number_of_maker_entries; i++){		
maker_ID	16	uimsbf
maker_model_code	16	uimsbf
start_mpd_block_number	16	uimsbf
reserved	16	bslbf
mpd_length	32	uimsbf
}		
stuffing_bytes	8*2*L1	bslbf
for(j=0; j<number_of_mpd_blocks; j++){		
mpd_block	mpd_block_size*1024*8	
}		
}		
}		

FIG.22

21/93

シンタクス	バイト数	略号
xxxxx.rpls / yyyy.ypls {		
PlayListMark_Start_address	32	uimsbf
MakersPrivateData_Start_address	32	uimsbf
reserved	192	bslbf
PlayList()		
for (i=0;i<N1;i++){		
padding_word	16	bslbf
}		
PlayListMark()		
for (i=0;i<N2;i++){		
padding_word	16	bslbf
}		
MakersPrivateData()		
}		

FIG.23

22/93

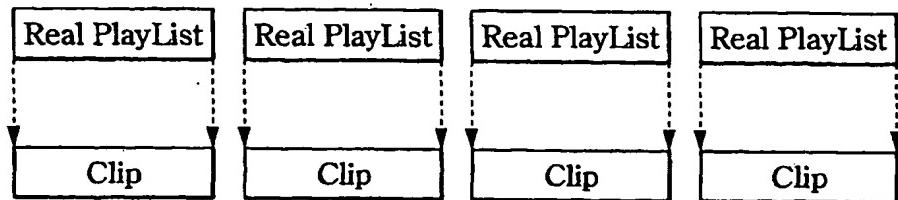


FIG.24A

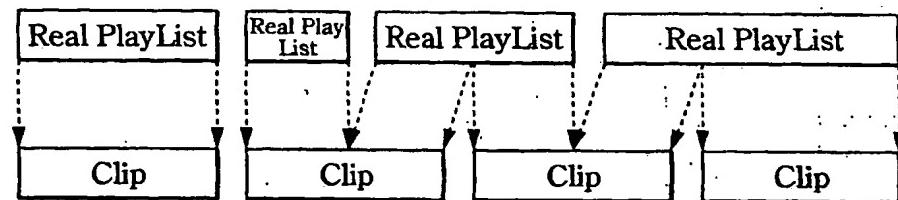


FIG.24B

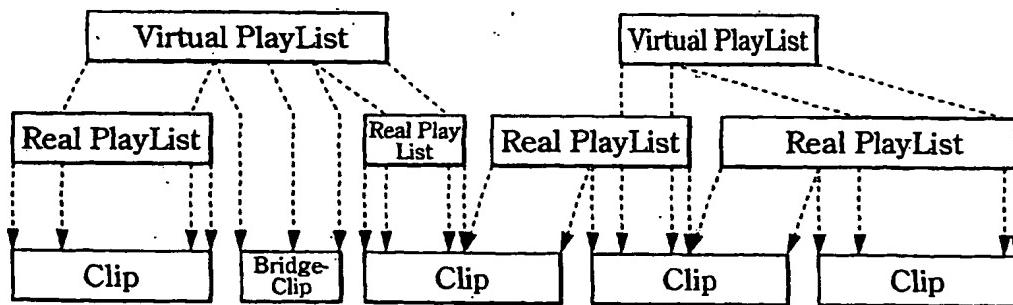


FIG.24C

23/93

シンタクス	バイト数	略号
PlayList()		
version_number	8*4	bslbf
length	32	uimsbf
PlayList_type	8	uimsbf
CPI_type	1	bslbf
reserved	7	bslbf
UIAppInfoPlayList()		
number_of_PlayItems // main path	16	uimsbf
if (<Virtual PlayList>){		
number_of_SubPlayItems // sub path	16	uimsbf
}else{		
reserved	16	bslbf
}		
for (PlayItem_id=0;		
PlayItem_id<number_of_PlayItems;		
PlayItem_id++){		
PlayItem() //main path		
}		
if (<Virtual PlayList>){		
if (CPI_type==0 && PlayList_type==0){		
for (i=0; i<number_of_SubPlayItems; i++)		
SubPlayItem() //sub path		
}		
}		
}		

FIG.25

24/93

PlayList_type	意味
0	AV記録のためのPlayList このPlayListに参照されるすべてのClipは、一つ以上のビデオストリームを含まなければならない。
1	オーディオ記録のためのPlayList このPlayListに参照されるすべてのClipは、一つ以上のオーディオストリームを含まなければならない、そしてビデオストリームを含んではならない。
2-255	reserved

FIG.26

25/93

シンタクス	バイト数	略号
UIAppInfoPlayList20{		
character_set	8	bslbf
name_length	8	uimsbf
PlayList_name	8*256	bslbf
reserved	8	bslbf
record_time_and_date	4*14	bslbf
reserved	8	bslbf
duration	4*6	bslbf
valid_period	4*8	bslbf
maker_id	16	uimsbf
maker_code	16	uimsbf
reserved	11	bslbf
playback_control_flag	1	bslbf
write_protect_flag	1	bslbf
is_played_flag	1	bslbf
archive	2	bslbf
ref_thumbnail_index	16	uimsbf
reserved_for_future_use	256	bslbf
}		

FIG.27

26/93

write_protect_flag	意味
0b	そのPlayListを自由に消去しても良い。
1b	write_protect_flagを除いてそのPlayListの内容は、消去および変更されるべきではない。

FIG.28A

is_played_flag	意味
0b	そのPlayListは、記録されてから一度も再生されたことがない。
1b	PlayListは、記録されてから一度は再生された。

FIG.28B

archive	意味
00b	何も情報が定義されていない。
01b	オリジナル
10b	コピー
11b	reserved

FIG.28C

27/93

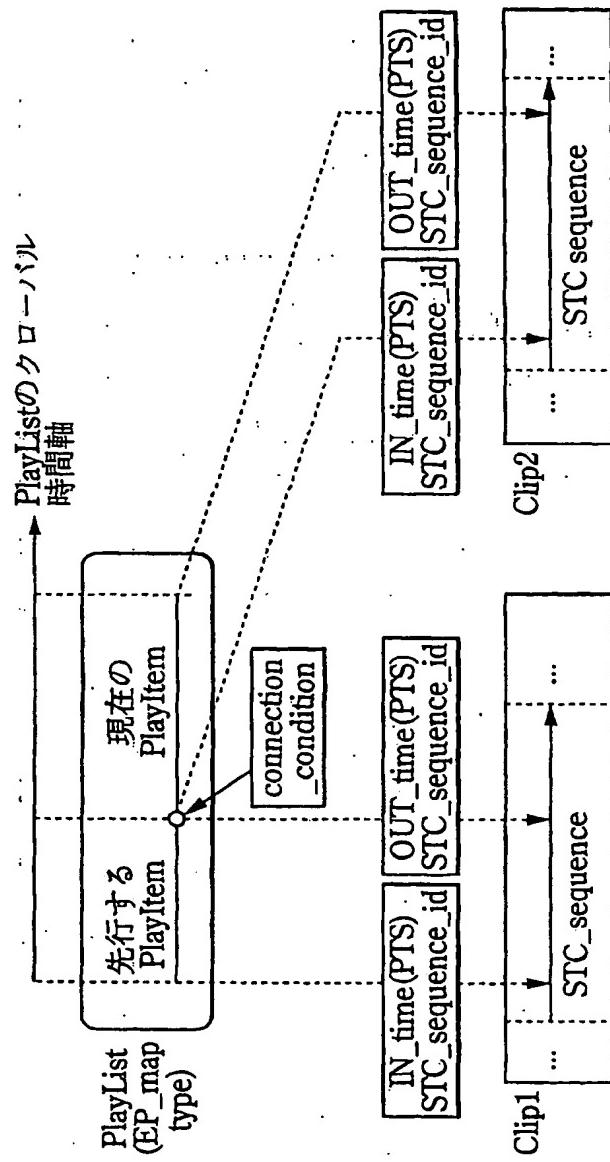


FIG.29

28/93

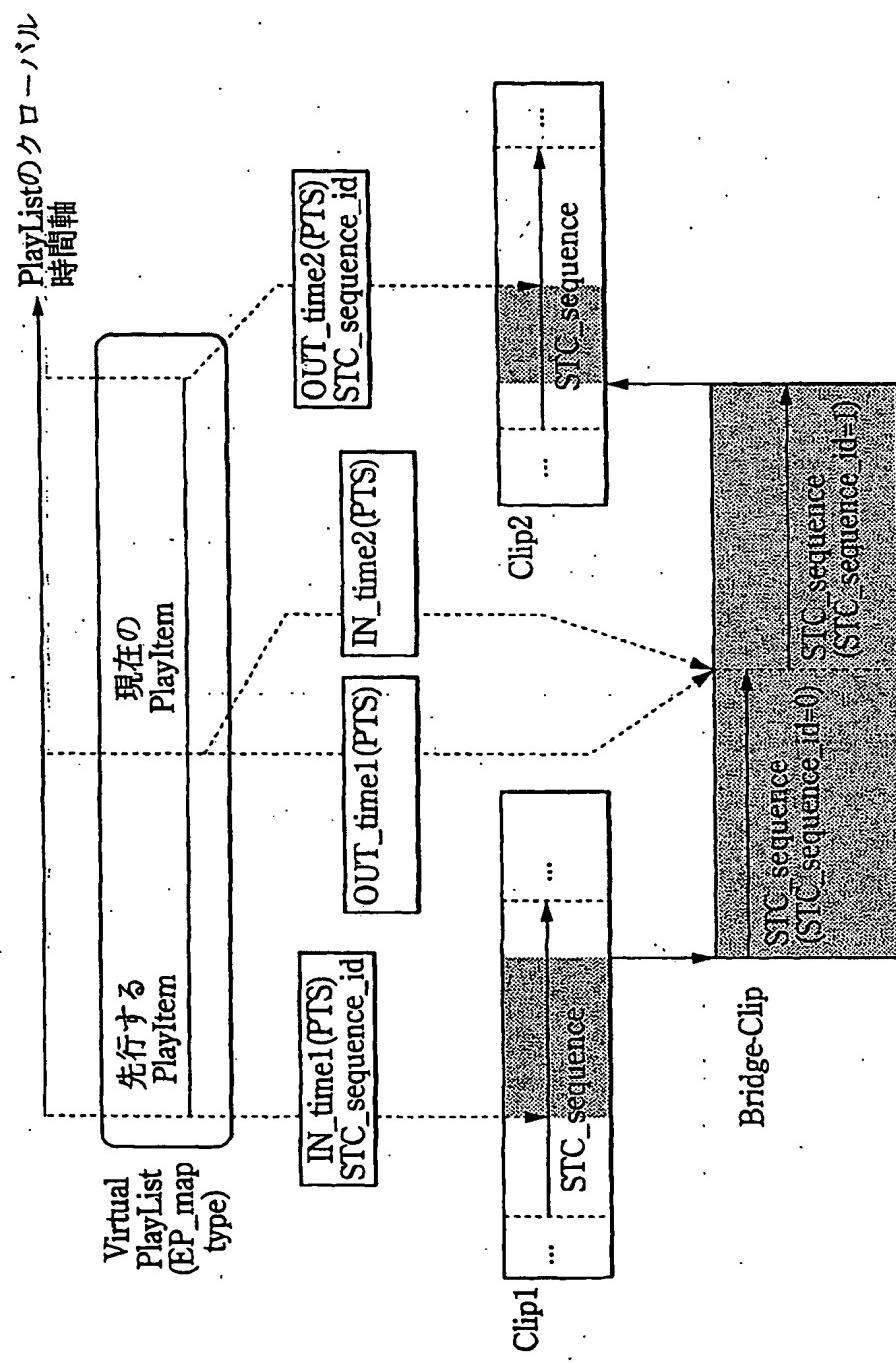


FIG.30

29/93

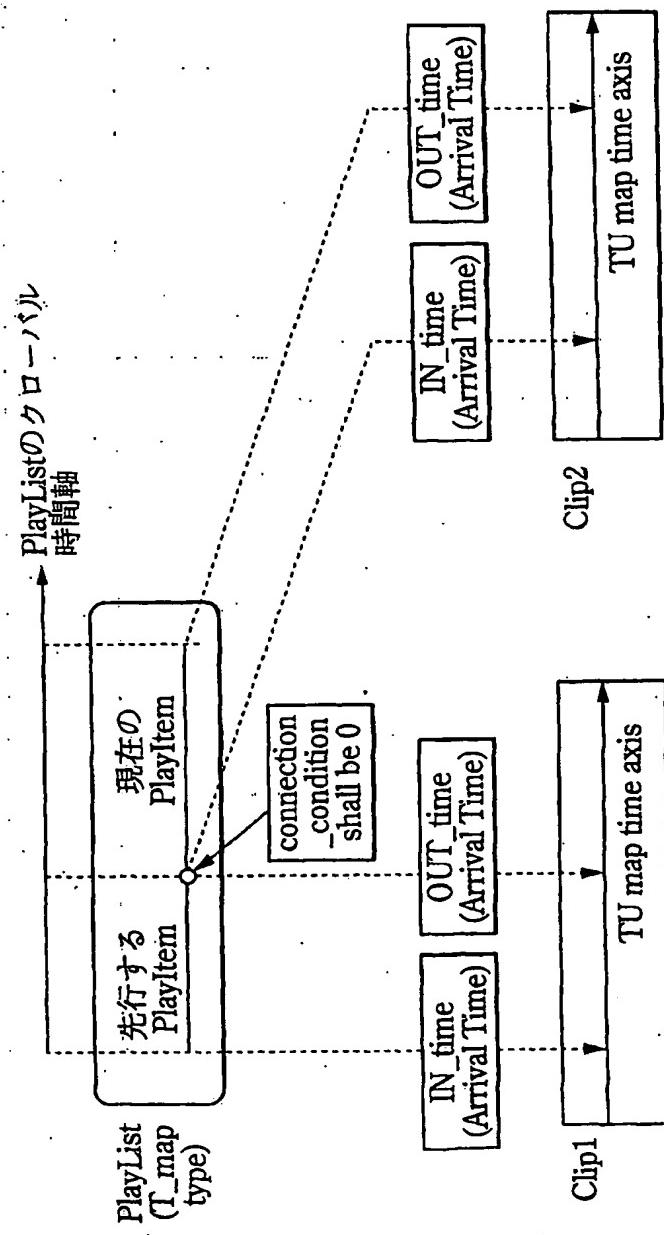


FIG.31

30/93

シンタクス	バイト数	略号
PlayItem()		
Clip_information_file_name	8*10	bslbf
reserved	24	bslbf
STC_sequence_id	8	uimsbf
IN_time	32	uimsbf
OUT_time	32	uimsbf
reserved	14	bslbf
connection_condition	2	bslbf
if (<Virtual PlayList>){		
if (connection_condition=='10')		
BridgeSequenceInfo()		
}		
}		
}		

FIG.32

31/93

CPI_type in the PlayList()	IN_timeのセマンティクス
EP_map type	IN_timeは、PlayItemの中で最初のプレゼンテーションユニットに対応する33ビット長のPTSの上位32ビットを示さなければならない。
TU_map type	IN_timeは、TU_map_time_axis上の時刻でなければならない。かつ、IN_timeは、time_unitの精度に丸めて表さなければならない。IN_timeは、次に示す等式により計算される。 $\text{IN_time} = \text{TU_start_time \%}2^{32}$

FIG.33

32/93

CPI_type in the Playlist()	OUT_timeのセマンティクス
EP_map type	<p>OUT_timeは、次に示す等式により計算される Presentation_end_TSの値の上位32ビットを示さなければならない。</p> $\text{Presentation_end_TS} = \text{PTS_out} + \text{AU_duration}$ <p>ここで、</p> <p>PTS_outは、PlayItemの中で最後のプレゼンテーションユニットに対応する33ビット長のPTSである。</p> <p>AU_durationは、最後のプレゼンテーションユニットの90kHz単位の表示期間である。</p>
TU_map type	<p>OUT_timeは、TU_map_time_axis上の時刻でなければならない。かつ、OUT_timeは、time_unitの精度に丸めて表さなければならない。OUT_timeは、次に示す等式により計算される。</p> $\text{OUT_time} = \text{TU_start_time \%}2^{32}$

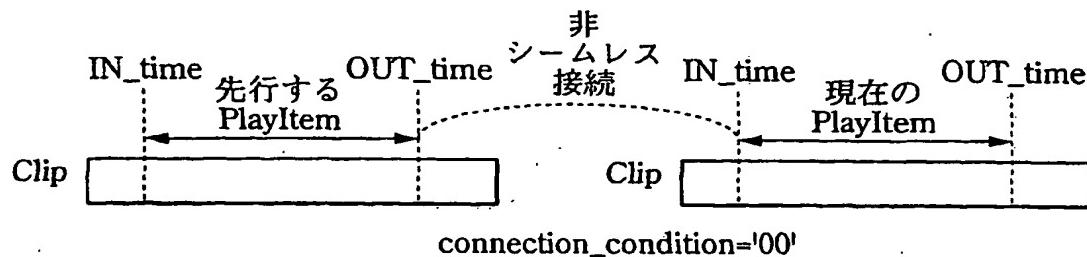
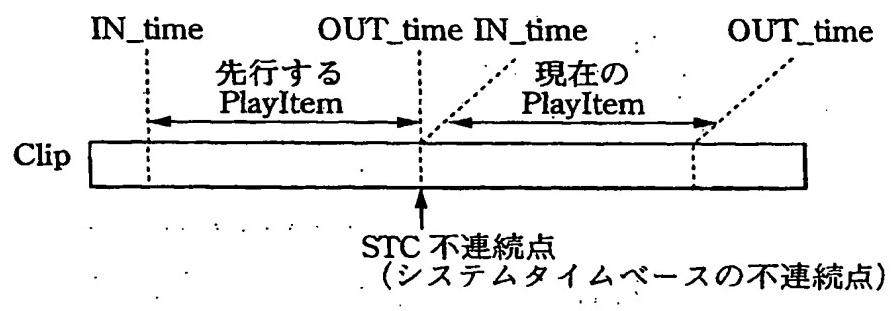
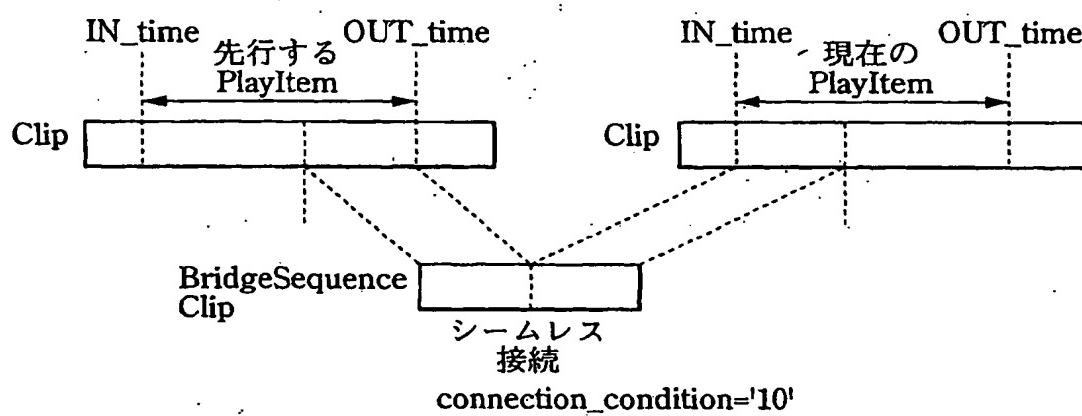
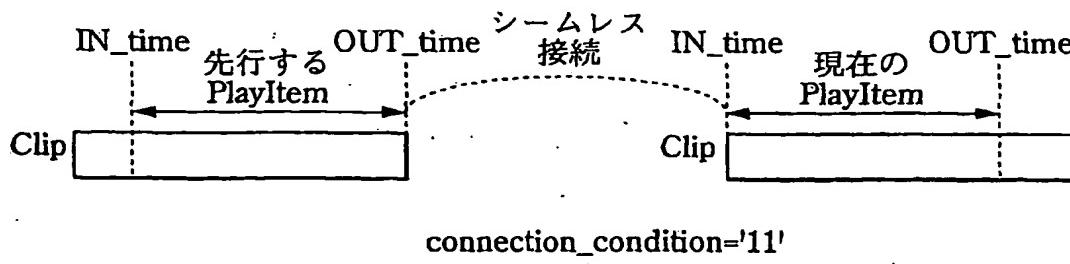
FIG.34

33/93

connection _condition	意味
00	<ul style="list-style-type: none"> 先行するPlayItemと現在のPlayItemの接続は、シームレス再生の保証がなされていない。 PlayListのCPI_typeがTU_map_typeである場合、connection_conditionは、この値をセットされねばならない。
01	<ul style="list-style-type: none"> この状態は、PlayListのCPI_typeがEP_map_typeである場合にだけ許される。 先行するPlayItemと現在のPlayItemは、システムタイムベース(STCベース)の不連続点があるために分割されていることを表す。
10	<ul style="list-style-type: none"> この状態は、PlayListのCPI_typeがEP_map_typeである場合にだけ許される。 この状態は、Virtual PlayListに対してだけ許される。 先行するPlayItemと現在のPlayItemとの接続は、シームレス再生の保証がなされている。 先行するPlayItemと現在のPlayItemは、BridgeSequenceを使用して接続されており、DVR MPEG-2トランスポートストリームは、後述するDVR-STDに従っていなければならぬ。
11	<ul style="list-style-type: none"> この状態は、PlayListのCPI_typeがEP_map_typeである場合にだけ許される。 先行するPlayItemと現在のPlayItemは、シームレス再生の保証がなされている。 先行するPlayItemと現在のPlayItemは、BridgeSequenceを使用しないで接続されており、DVR MPEG-2トランスポートストリームは、後述するDVR-STDに従っていなければならぬ。

FIG.35

34/93

**FIG.36A****FIG.36B****FIG.36C****FIG.36D**

35/93

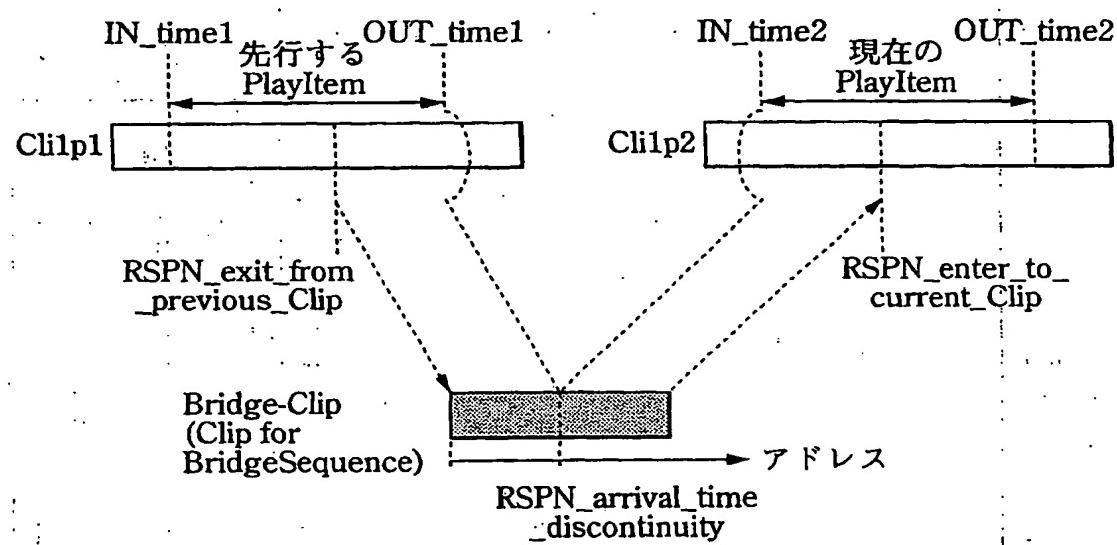


FIG.37

36/93

シンタクス	バイト数	略号
BridgeSequenceInfo()		
Bridge_Clip_information_file_name	8*10	bslbf
RSPN_exit_from_previous_Clip	32	uimsbf
RSPN_enter_to_current_Clip	32	uimsbf
}		

FIG.38

37/93

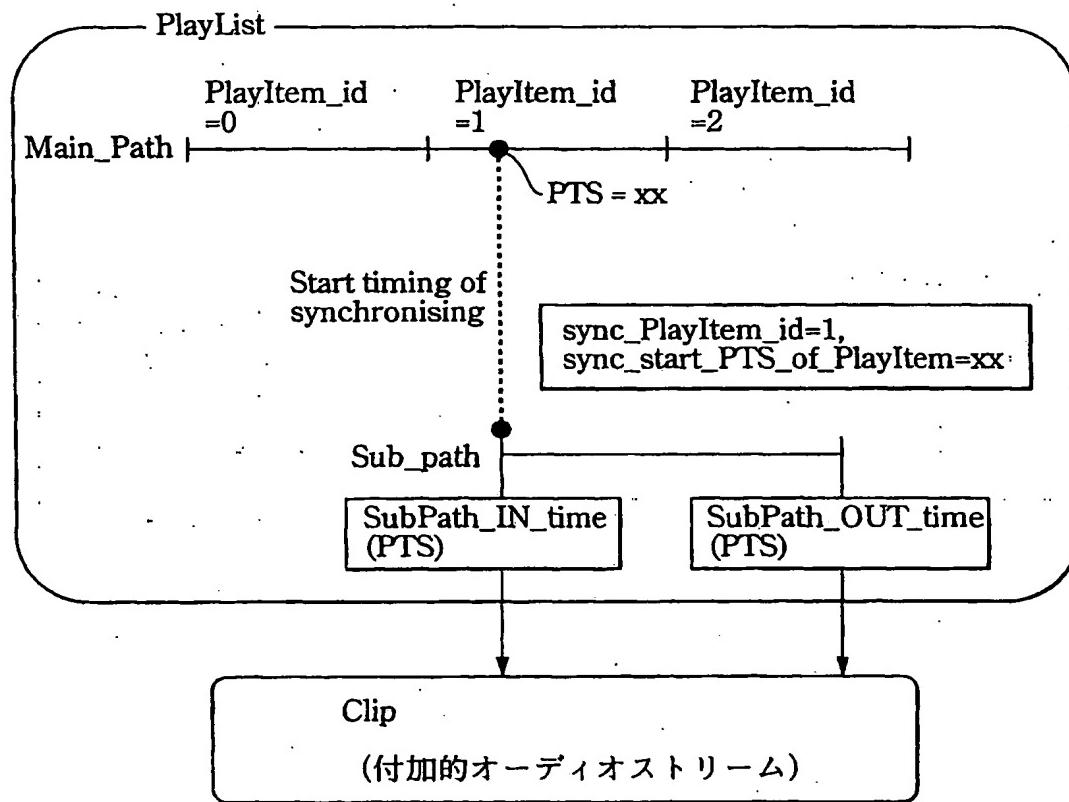


FIG.39

38/93

シンタクス	バイト数	略号
SubPlayItem01		
Clip_Information_file_name	8*10	bslbf
SubPath_type	8	bslbf
sync_PlayItem_id	8	uimsbf
sync_start PTS_of_PlayItem	32	uimsbf
SubPath_IN_time	32	uimsbf
SubPath_OUT_time	32	uimsbf
}		

FIG.40

39/93

SubPath_type	意味
0x00	Auxiliary audio steam path
0x01-0xff	reserved

FIG.41

40/93

シンタクス	バイト数	略号
PlayListMark(){		
version_number	8*4	bslbf
length	32	uimsbf
number_of_PlayList_marks	16	uimsbf
for (i=0;i<number_of_PlayList_marks;i++){		
reserved	8	bslbf
mark_type	8	bslbf
mark_time_stamp	32	uimsbf
PlayItem_id	8	uimsbf
reserved	24	uimsbf
character_set	8	bslbf
name_length	8	uimsbf
mark_name	8*256	bslbf
ref_thumbnail_index	16	uimsbf
}		
}		

FIG.42

41/93

Mark_type	意味	コメント
0x00	resume-mark	再生リジュームポイント。PlayListMark()において定義される再生リジュームポイントの数は、0または1でなければならない。
0x01	book-mark	PlayListの再生エントリーポイント。このマークは、ユーザがセットすることができ、例えば、お気に入りのシーンの開始点を指定するマークに使う。
0x02	skip-mark	スキップマークポイント。このポイントからプログラムの最後まで、プレーヤはプログラムをスキップする。PlayListMark()において定義されるスキップマークポイントの数は、0または1でなければならない。
0x03-0x8F	reserved	
0x90-0xFF	reserved	Reserved for ClipMark()

FIG.43

42/93

CPI_type in the PlayList()	mark_time_stampのセマンティクス
EP_map type	mark_time_stampは、マークで参照されるプレゼンテーションユニットに対応する33ビット長のPTSの上位32ビットを示さなければならない。
TU_map type	mark_time_stampは、TU_map_time_axis上の時刻でなければならない。かつ、mark_time_stampは、time_unitの精度に丸めて表さなければならない。mark_time_stampは、次に示す等式により計算される。 $\text{mark_time_stamp} = \text{TU_start_time \%}2^{32}$

FIG.44

43/93

シンタクス	バイト数	略号
zzzzz.clpi {		
STC_Info_Start_address	32	uimsbf
ProgramInfo_Start_address	32	uimsbf
CPI_Start_address	32	uimsbf
ClipMark_Start_address	32	uimsbf
MakersPrivateData_Start_address	32	uimsbf
reserved	96	bslbf
ClipInfo()		
for (i=0;i<N1;i++){		
padding_word	16	bslbf
}		
STC_Info()		
for (i=0;i<N2;i++){		
padding_word	16	bslbf
}		
ProgramInfo()		
for (i=0;i<N3;i++){		
padding_word	16	bslbf
}		
CPI()		
for (i=0;i<N4;i++){		
padding_word	16	bslbf
}		
ClipMark()		
for (i=0;i<N5;i++){		
padding_word	16	bslbf
}		
MakersPrivateData()		
}		

FIG.45

44/93

シンタクス	バイト数	略号
ClipInfo0{		
version_number	8*4	bslbf
length	32	uimsbf
Clip_stream_type	8	bslbf
offset_SPN	32	uimsbf
TS_recording_rate	24	uimsbf
reserved	8	bslbf
record_time_and_date	4*14	bslbf
reserved	8	bslbf
duration	4*6	bslbf
reserved	7	bslbf
time_controlled_flag	1	bslbf
TS_average_rate	24	uimsbf
if(Clip_stream_type==1) // Bridge-Clip AV stream		
RSPN_arrival_time_discontinuity	32	uimsbf
else		
reserved	32	bslbf
reserved_for_system_use	144	bslbf
reserved	11	bslbf
is_format_identifier_valid	1	bslbf
is_original_network_ID_valid	1	bslbf
is_transport_stream_ID_valid	1	bslbf
is_service_ID_valid	1	bslbf
is_country_code_valid	1	bslbf
format_identifier	32	bslbf
original_network_ID	16	uimsbf
transport_stream_ID	16	uimsbf
service_ID	16	uimsbf
country_code	24	bslbf
stream_format_name	16*8	bslbf
reserved_for_fortune_use	256	bslbf
}		

FIG.46

45/93

Clip_stream_type	意味
0	Clip AV ストリーム
1	Bridge-Clip AV ストリーム
2-255	Reserved

FIG.47

46/93

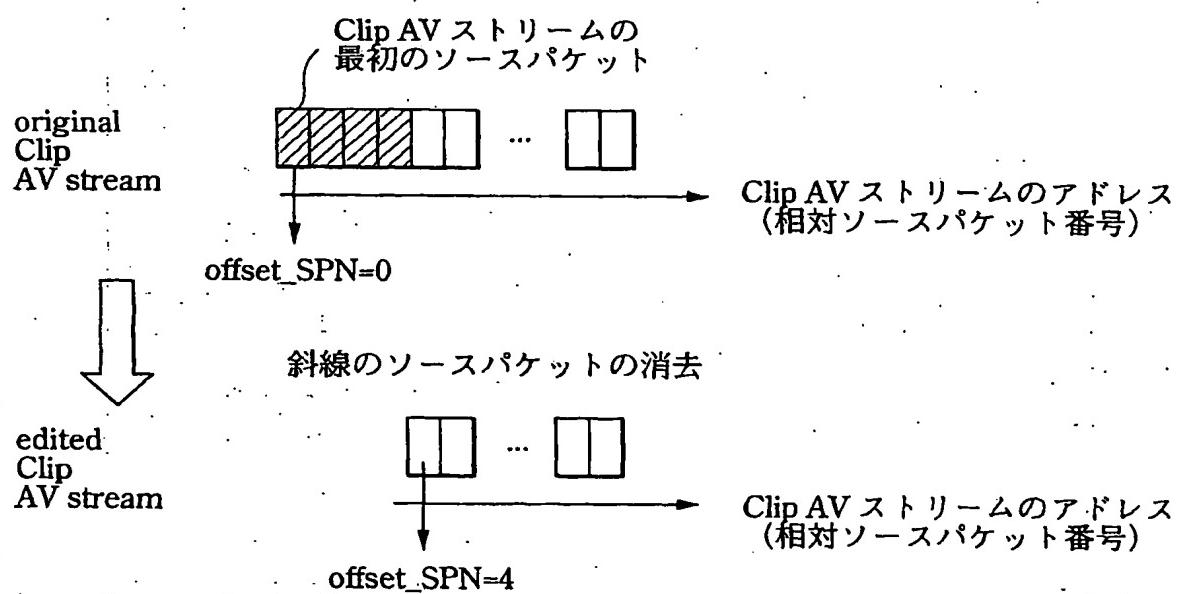
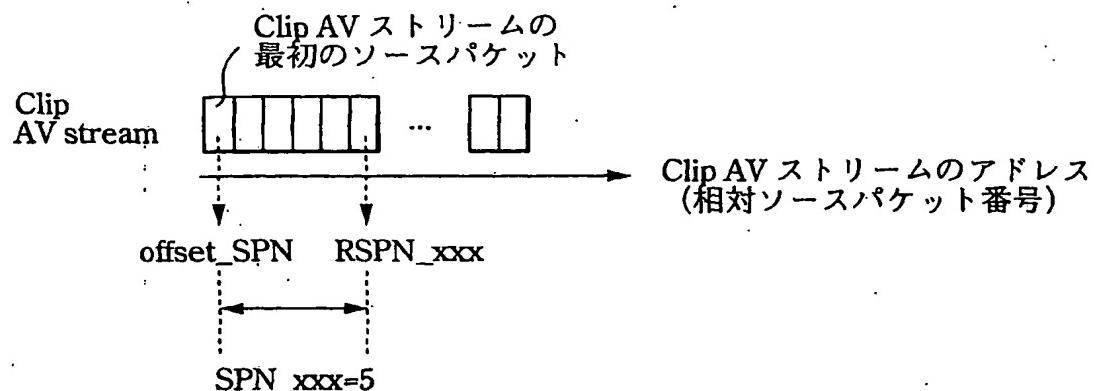
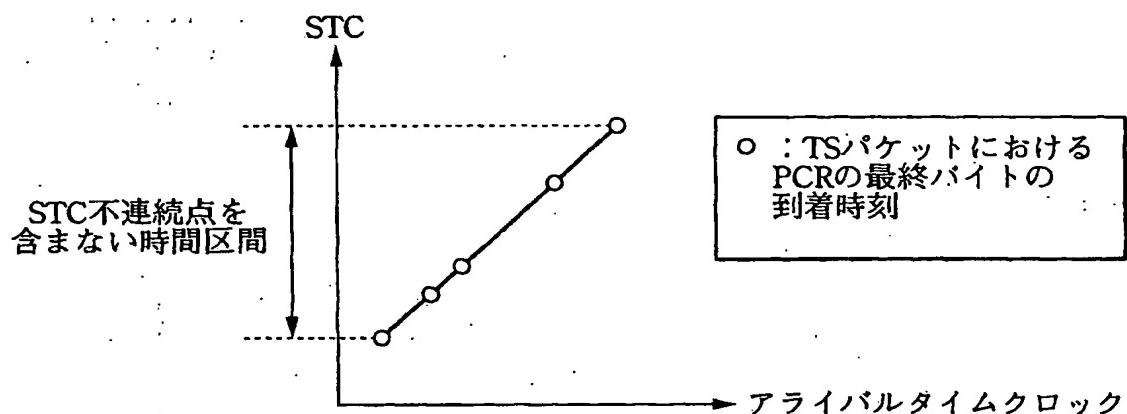
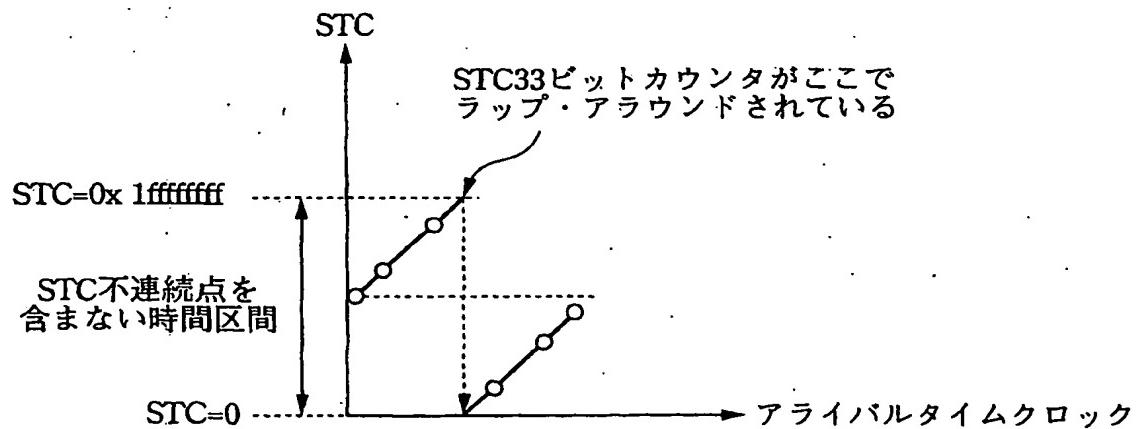
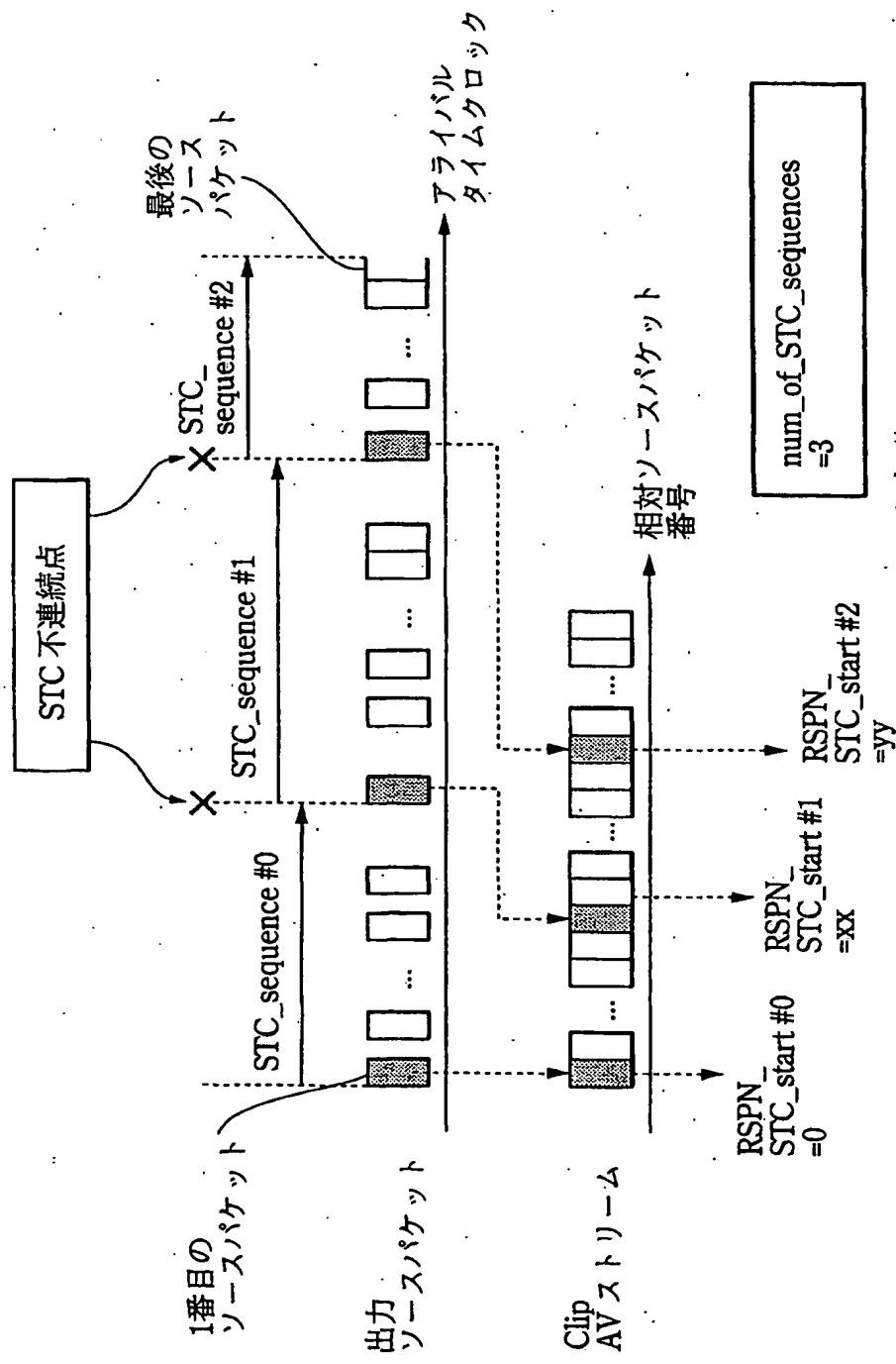


FIG.48

47/93

**FIG.49****FIG.50A****FIG.50B**

48/93

**FIG.51**

49/93

シンタクス	バイト数	略号
STC_Info0{		
version_number	8*4	bslbf
length	32	uimsbf
if (length !=0){		
reserved	8	bslbf
num_of_STC_sequences	8	uimsbf
for (STC_sequence_id=0;		
STC_sequence_id<num_of_STC_sequences;		
STC_sequence_id++){		
resereved	32	bslbf
RSPN_STC_start	32	uimsbf
}		
}		
}		

FIG.52

50/93

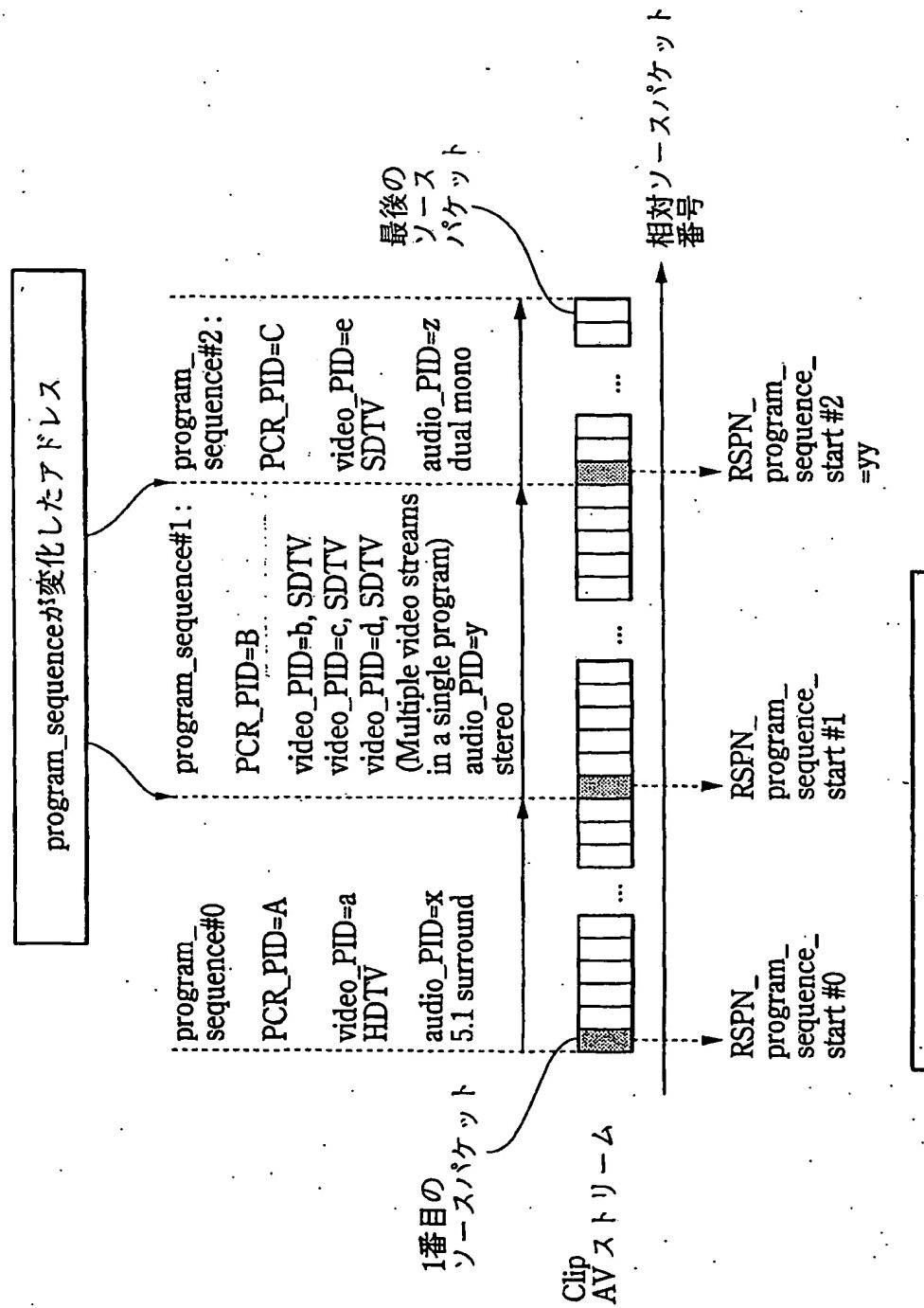


FIG.53

51/93

シンタクス	バイト数	略号
ProgramInfo()		
version_number	8*4	bslbf
length	32	uimsbf
if (length !=0){		
reserved	8	bslbf
number_of_program_sequences	8	uimsbf
for (i=0;i<number_of_program_sequences;i++){		
RSPN_program_sequence_start	32	uimsbf
reserved	48	bslbf
PCR_PID	16	bslbf
number_of_videos	8	uimsbf
number_of_audios	8	uimsbf
for (k=0;k<number_of_videos;k++){		
video_stream_PID	16	bslbf
VideoCodingInfo()		
}		
for (k=0;k<number_of_audios;k++){		
audio_stream_PID	16	bslbf
AudioCodingInfo()		
}		
}		
}		

FIG.54

52/93

シンタクス	バイト数	略号
VideoCodingInfo()		
video_format	8	uimsbf
frame_rate	8	uimsbf
display_aspect_ratio	8	uimsbf
reserved	8	bslbf
}		

FIG.55

53/93

video_format	意味
0	480i
1	576i
2	480p(including 640×480p format)
3	1080i
4	720p
5	1080p
6-254	reserved
255	No information

FIG.56

frame_rate	意味
0	forbidden
1	24 000/1001 (23.976...)
2	24
3	25
4	30 000/1001 (29.97..)
5	30
6	50
7	60 000/1001 (59.94..)
8	60
9-254	reserved
255	No information

FIG.57

54/93

display_aspect_ratio	意味
0	forbidden
1	reserved
2	4:3 display aspect ratio
3	16:9 display aspect ration
4-254	reserved
255	No information

FIG.58

55/93

シンタクス	バイト数	略号
AudioCodingInfo()		
audio_format	8	uimsbf
audio_component_type	8	uimsbf
sampling_frequency	8	uimsbf
reserved	8	bslbf
}		

FIG.59

56/93

audio_coding	意味
0	MPEG-1 audio layer I or II
1	Dolby AC-3 audio
2	MPEG-2 AAC
3	MPEG-2 multi-channel audio, backward compatible to MPEG-1
4	SESF LPCM audio
5-254	reserved
255	No information

FIG.60

57/93

audio_component_type	意味
0	single mono channel
1	dual mono channel
2	stereo (2-channel)
3	multi-lingual, multi-channel
4	surround sound
5	audio description for the visually impaired
6	audio for the hard of hearing
7-254	reserved
255	No information

FIG.61

sampling_frequency	意味
0	48 kHz
1	44.1 kHz
2	32 kHz
3-254	reserved
255	No information

FIG.62

58/93

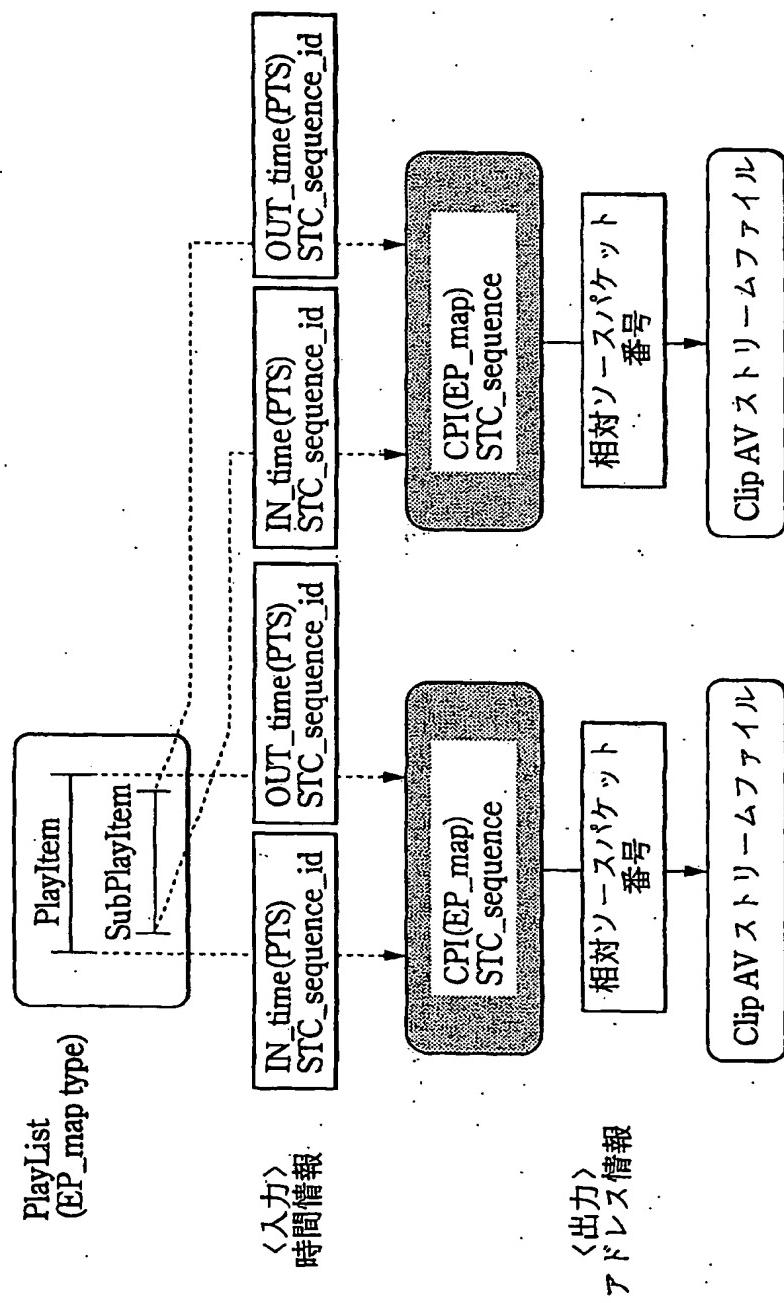


FIG.63

59/93

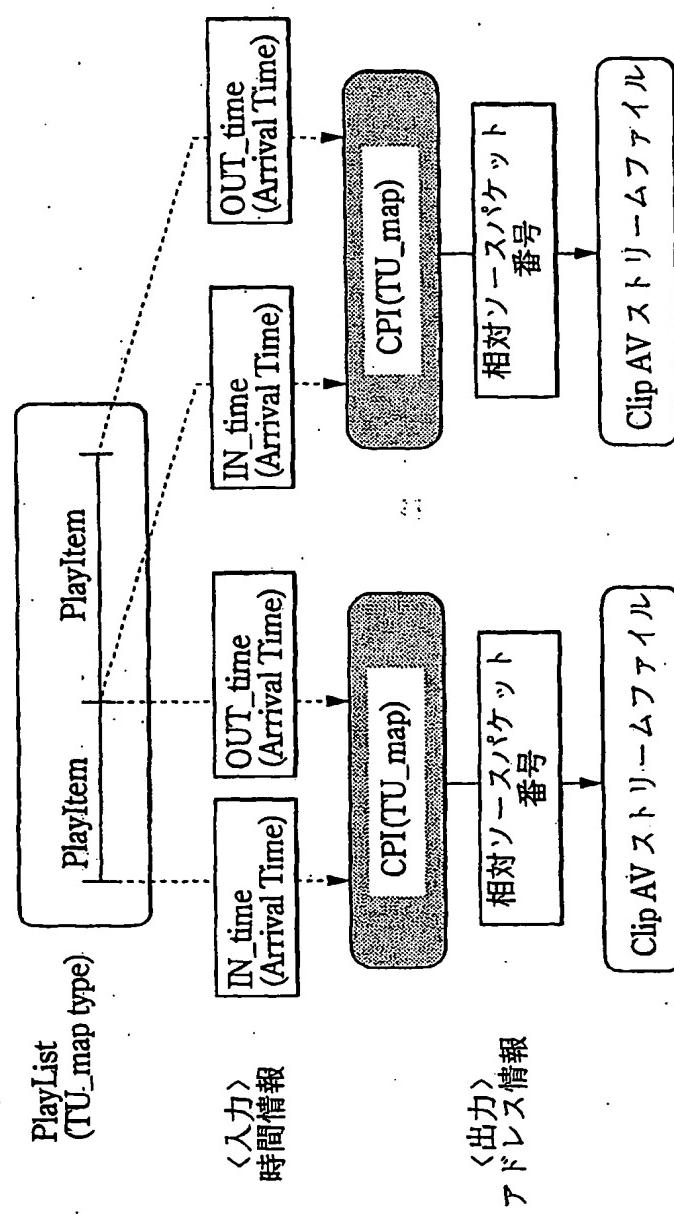


FIG.64

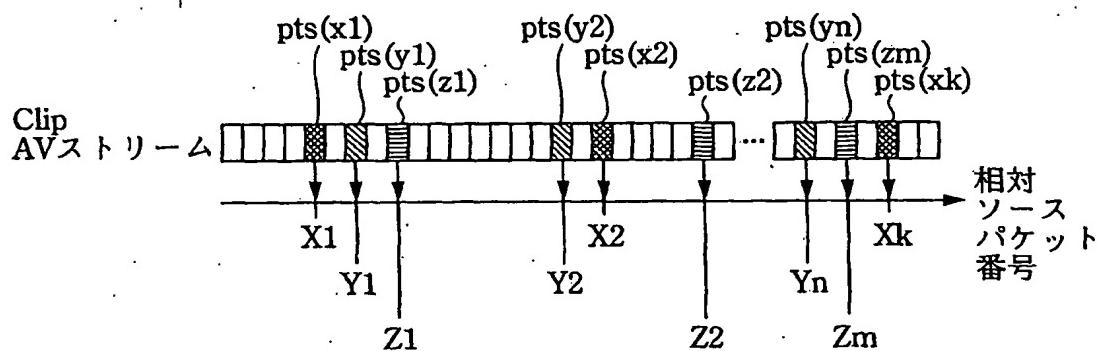
60/93

シンタクス	バイト数	略号
CPI{		
version_number	8*4	bslbf
length	32	uimsbf
reserved	15	bslbf
CPI_type	1	bslbf
if (CPI_type==0)		
EP_map()		
else		
TU_map()		
}		

FIG.65

61/93

CPI_type	意味
0	EP map type
1	TU map type

FIG.66**EP_map**

number_of_stream/PIDs=3

EP_map_for_one_stream_PID(0)

stream_PID(0)=x	num_EP_entries(0)=k
PTS_EP_start	RSPN_EP_start
pts(x1)	X1
pts(x2)	X2
...	...
pts(xk)	Xk

EP_map_for_one_stream_PID(1)

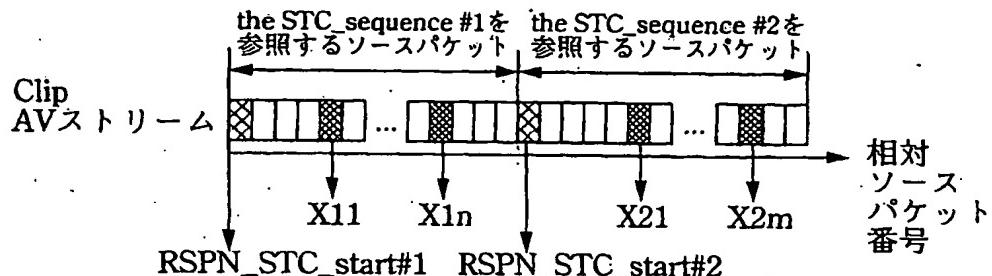
stream_PID(1)=y	num_EP_entries(1)=n
PTS_EP_start	RSPN_EP_start
pts(y1)	Y1
pts(y2)	Y2
...	...
pts(yn)	Yn

EP_map_for_one_stream_PID(2)

stream_PID(2)=z	num_EP_entries(2)=m
PTS_EP_start	RSPN_EP_start
pts(z1)	Z1
pts(z2)	Z2
...	...
pts(zm)	Zm

FIG.67

62/93



- : シーケンスヘッダvideo_PID=xの第1バイト目を含むソースパケット
- ▨ : RSPN_STC_startによって参照されるソースパケット(the STC_intoにおいて定義される)

EP_map_for_one_steram_PID
video_PID=x

PTS_EP_start	RSPN_EP_start	
pts(x11)	X11) STC_sequence #1に
...	X1n	属するデータ
pts(x21)	X21) STC_sequence #2に
...	X2m	属するデータ

boundary

RSPN_STC_start #2 < X21

FIG.68

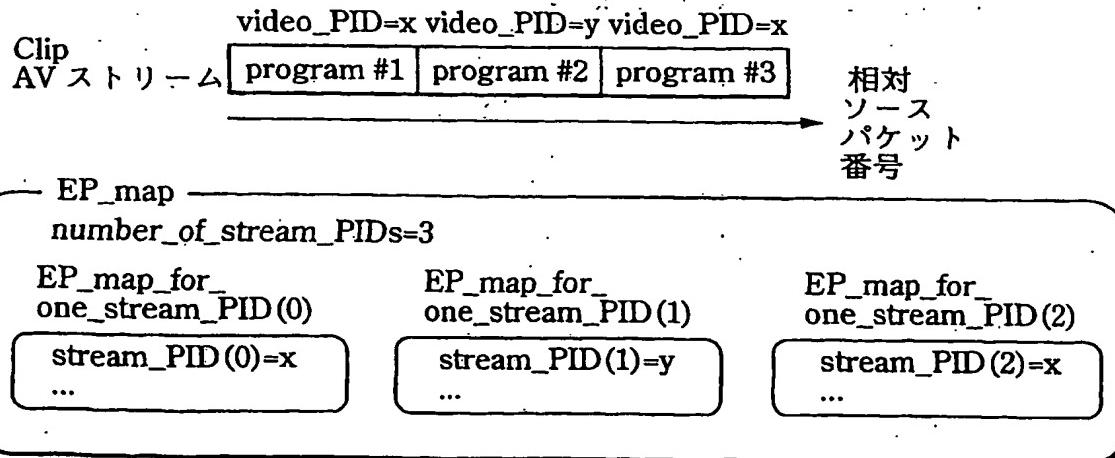


FIG.69

63/93

シンタクス	バイト数	略号
EP_map()		
reserved	12	bslbf
EP_type	4	uimsbf
number_of_stream_PIDs	16	uimsbf
for (k=0;k<number_of_stream_PIDs;k++){		
stream_PID(k)	16	bslbf
num_EP_entries(k)	32	uimsbf
EP_map_for_one_stream_PID_Start_address(k)	32	uimsbf
}		
for (i=0;i<X;i++){		
padding_word	16	bslbf
}		
for (k=0;k<number_of_stream_PIDs;k++){		
EP_map_for_one_stream_PID(num_EP_entries(k))		
for (i=0;i<Y;i++){		
padding_word	16	bslbf
}		
}		
}		

FIG.70

64/93

EP_type	意味
0	video
1	audio
2-15	reserved

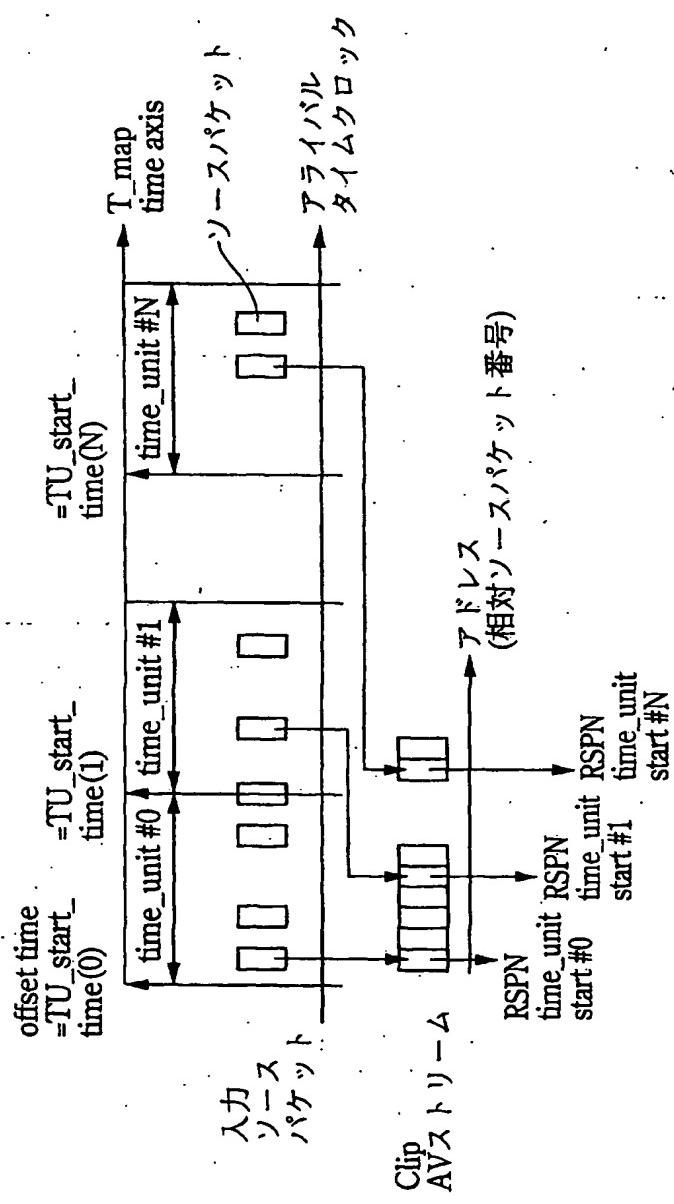
FIG.71

65/93

シンタクス	バイト数	略号
EP_map_for_one_stream_PID(<i>N</i>) {		
for (<i>i</i> =0; <i>i</i> < <i>N</i> ; <i>i</i> ++) {		
PTS_EP_start	32	uimsbf
RSPN_EP_start	32	uirmsbf
}		
}		

FIG.72

66/93

**FIG.73**

67/93

シンタクス	バイト数	略号
TU_map()		
offset_time	32	bslbf
time_unit_size	32	uimsbf
number_of_time_unit_entries	32	uimsbf
for (k=0;k<number_of_time_unit_entries;k++)		
RSPN_time_unit_start	32	uimsbf
}		

FIG.74

68/93

シンタクス	バイト数	略号
ClipMark {		
version_number	8*4	bslbf
length	32	uimsbf
number_of_Clip_marks	16	uimsbf
for (i=0; i<number_of_clip_marks; i++) {		
reserved	8	bslbf
mark_type	8	bslbf
mark_time_stamp	32	uimsbf
STC_sequence_id	8	uimsbf
reserved	24	bslbf
character_set	8	bslbf
name_length	8	uimsbf
mark_name	8*256	bslbf
ref_thumbnail_index	16	uimsbf
}		
}		

FIG.75

69/93

Mark_type	意味	コメント
0x00-0x8F	reserved	Reserved for PlayListMark()
0x90	Event-start mark	番組の開始ポイントを示すマーク点
0x91	Local event-start mark	番組の中の局所的な場面を示すマーク点
0x92	Scene-start mark	シーンチェンジポイントを示すマーク点
0x93-0xFF	reserved	

FIG.76

70/93

CPI_type in the CPI0	mark_time_stampのセマンティクス
EP_map type	mark_time_stampは、マークで参照されるプレゼンテーションユニットに対応する33ビット長のPTSの上位32ビットを示さなければならない。
TU_map type	mark_time_stampは、TU_map_time_axis上の時刻でなければならない。かつ、mark_time_stampは、time_unitの精度に丸めて表さなければならない。mark_time_stampは、次に示す等式により計算される。 $\text{mark_time_stamp} = \text{TU_start_time \%}2^{32}$

FIG.77

71/93

シンタクス	バイト数	略号
menu.thmb/mark.thmb {		
reserved	256	bslbf
Thumbnail()		
for (i=0; i<N1; i++)		
padding_word	16	bslbf
}		

FIG.78

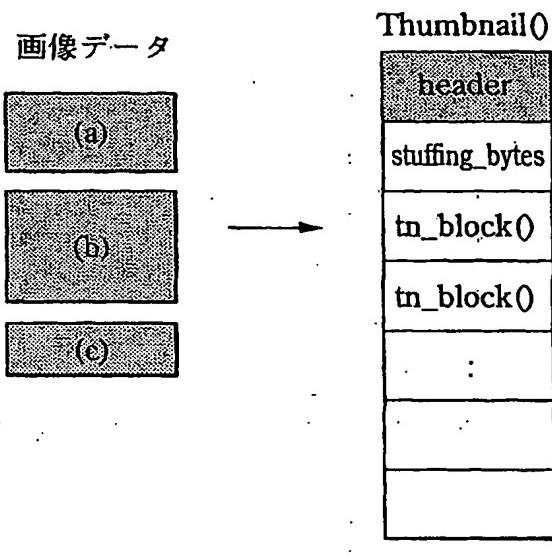
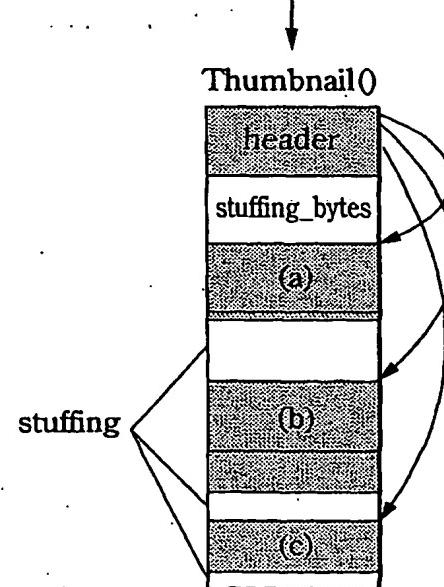
72/93

シンタクス	バイト数	略号
Thumbnail()		
version_number	8*4	char
length	32	uimsbf
if (length !=0){		
tn_blocks_start_address	32	bslbf
number_of_thumbnails	16	uimsbf
tn_block_size	16	uimsbf
number_of_tn_blocks	16	uimsbf
reserved	16	bslbf
for (i=0; i<number_of_thumbnails; i++){		
thumbnail_index	16	uimsbf
thumbnail_picture_format	8	bslbf
reserved	8	bslbf
picture_data_size	32	uimsbf
start_tn_block_number	16	uimsbf
x_picture_length	16	uimsbf
y_picture_length	16	uimsbf
reserved	16	uimsbf
}		
stuffing_bytes	8*2*L1	bslbf
for(k=0; k<number_of_tn_blocks; k++){		
tn_block	tn_block_size*1024*8	
}		
}		

FIG.79

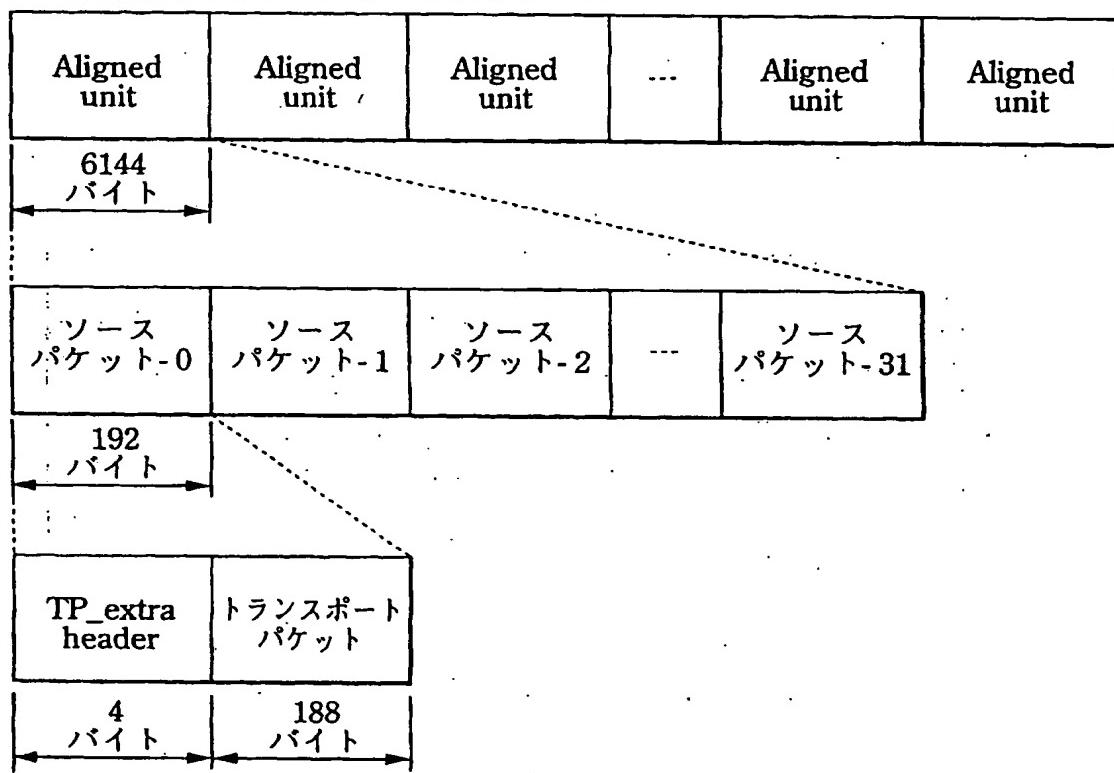
73/93

Thumbnail_picture_format	意味
0x00	MPEG-2 Video I-picture
0x01	DCF (restricted JPEG)
0x02	PNG
0x03-0xff	reserved

FIG.80**FIG.81A****FIG.81 B**

74/93

DVR MPEG-2 トランスポートストリーム

**FIG.82**

75/93

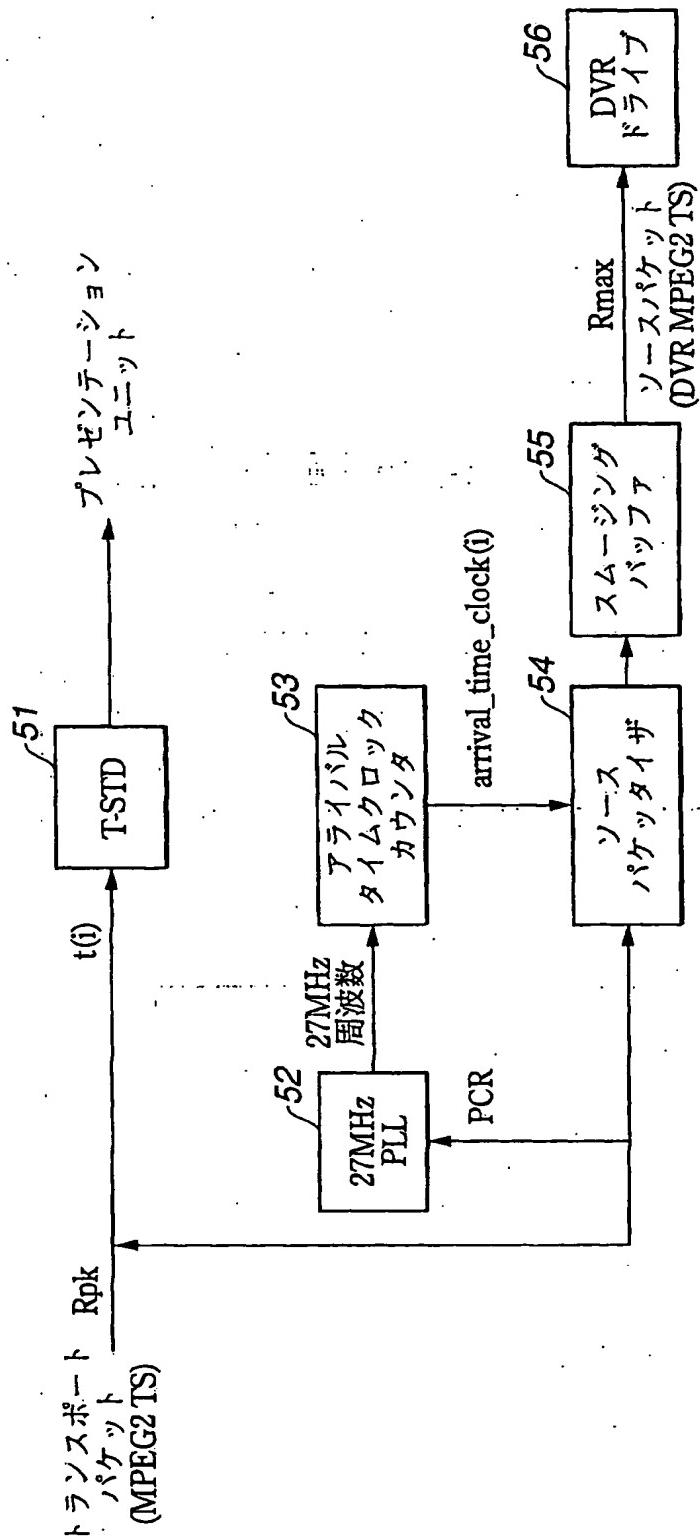


FIG.83

76/93

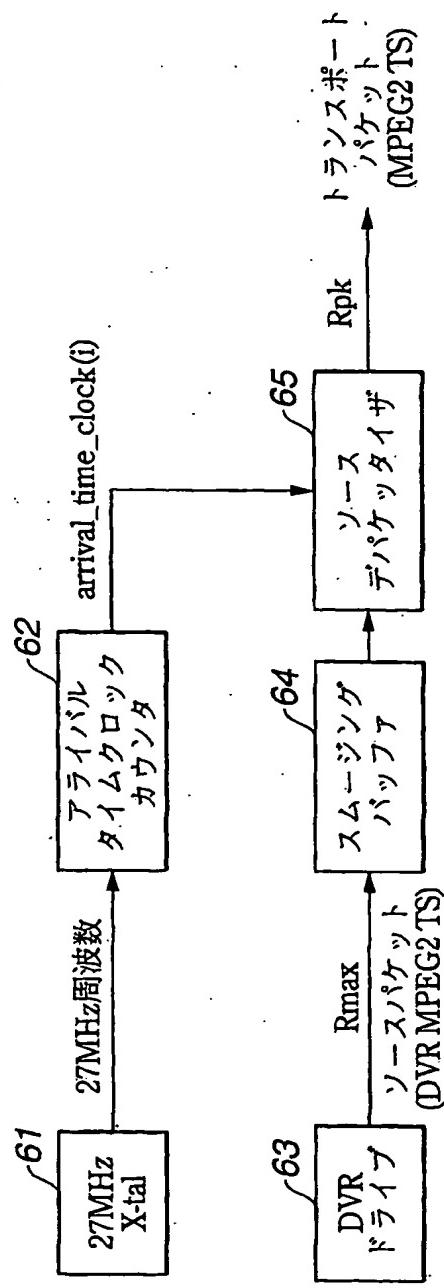


FIG.84

77/93

シンタクス	バイト数	略号
source_packet()		
TP_extra_header()		
transport_packet()		
}		

FIG.85

78/93

シンタクス	バイト数	略号
TP_extra_header()		
copy_permission_indicator	2	uimsbf
arrival_time_stamp	30	uimsbf
}		

FIG.86

79/93

copy_permission indicator	意味
00	copy free
01	no more copy
10	copy once
11	copy prohibited

FIG.87

80/93

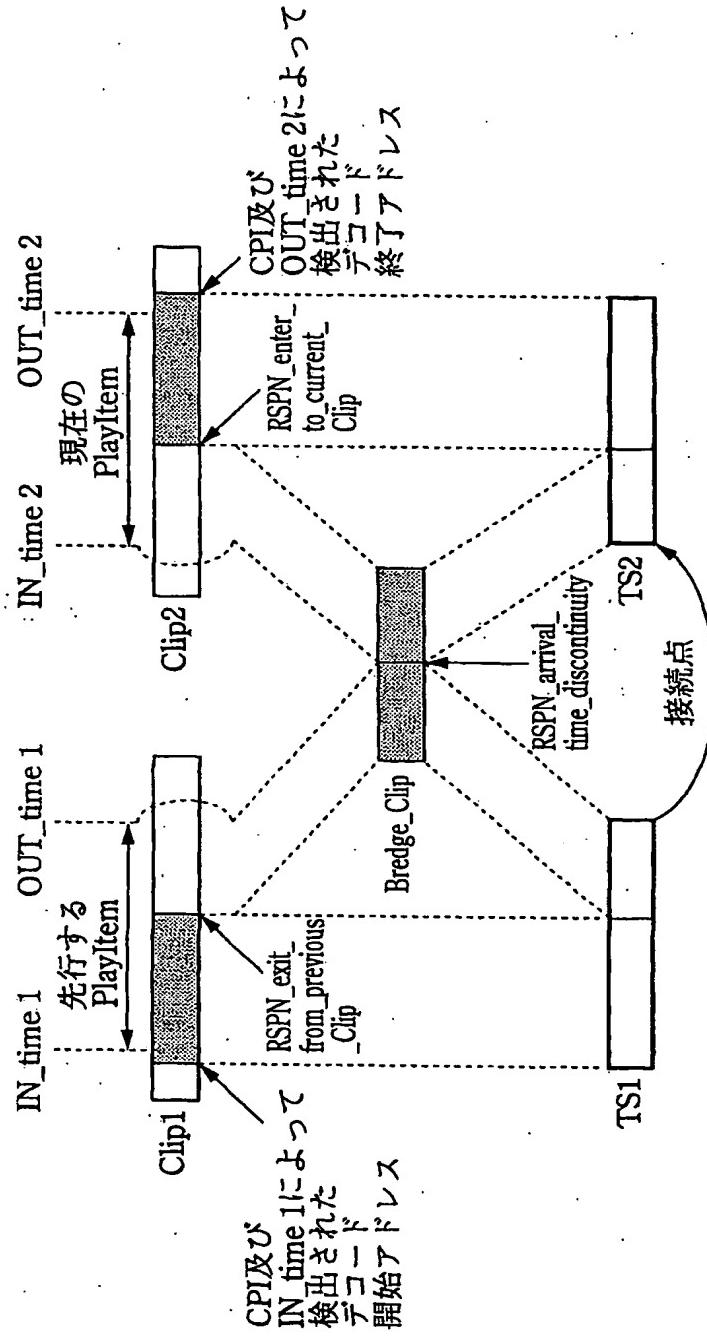


FIG.88

81/93

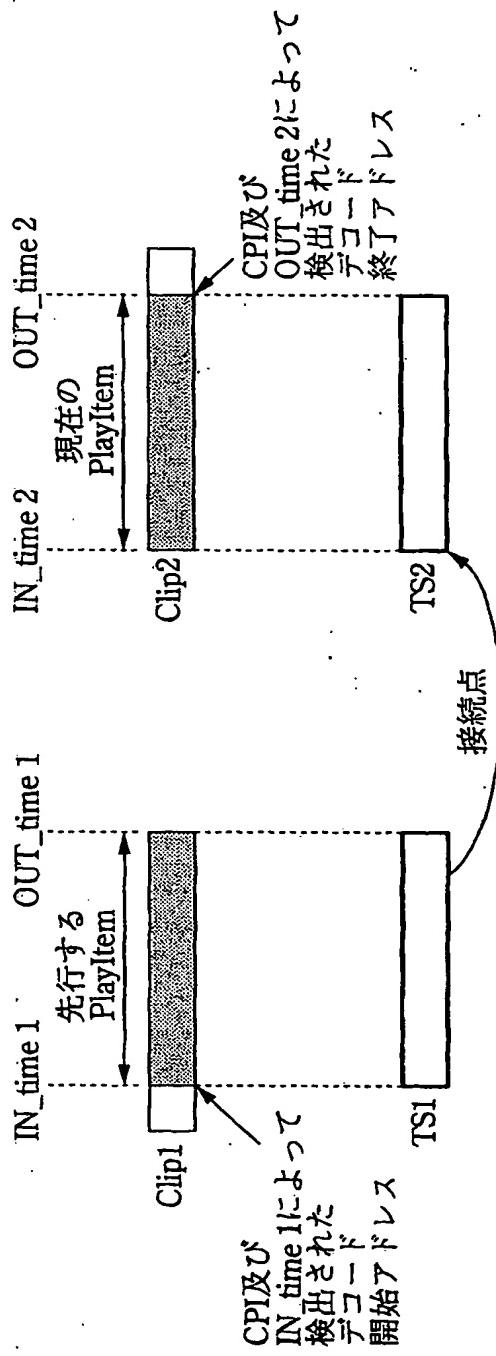


FIG.89

82/93

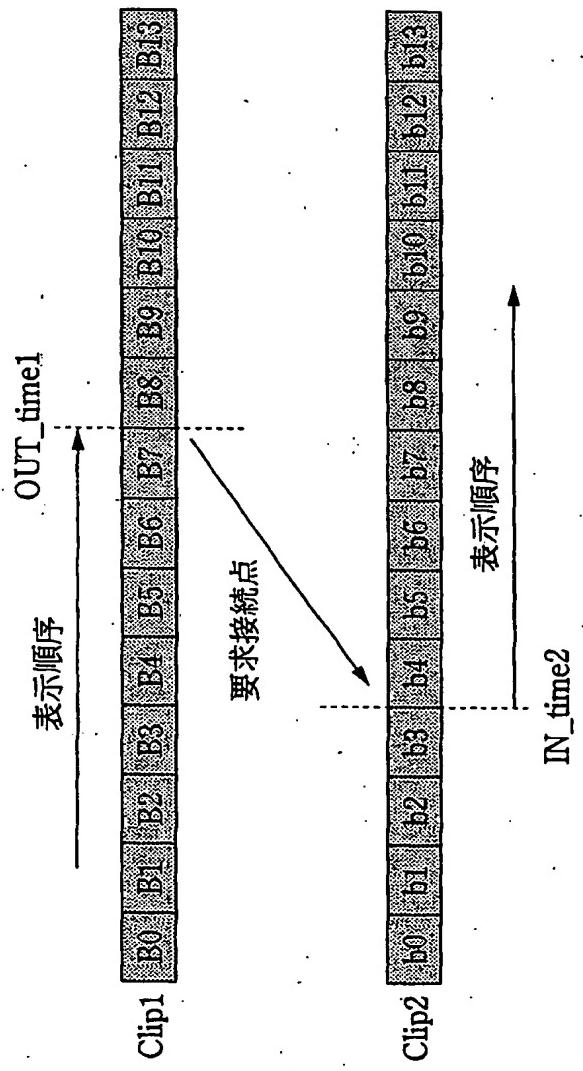
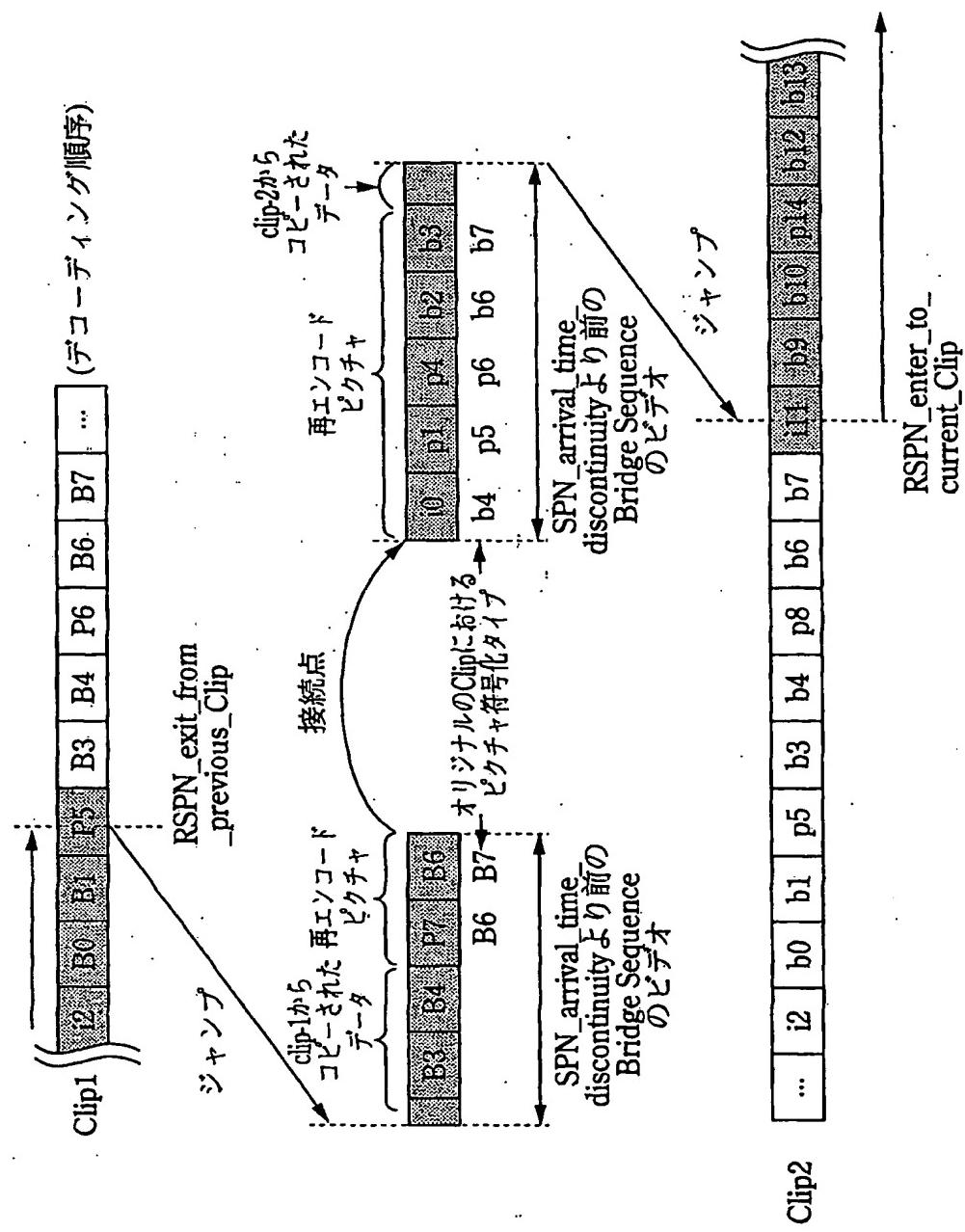
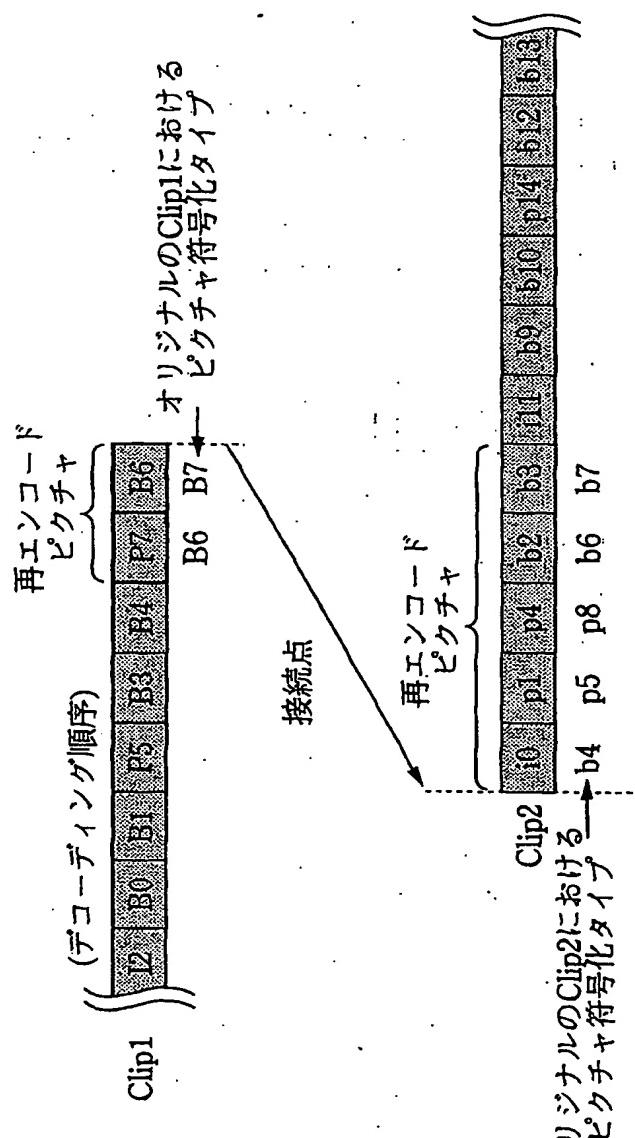


FIG.90

83/93

**FIG.91**

84/93

**FIG.92**

85/93

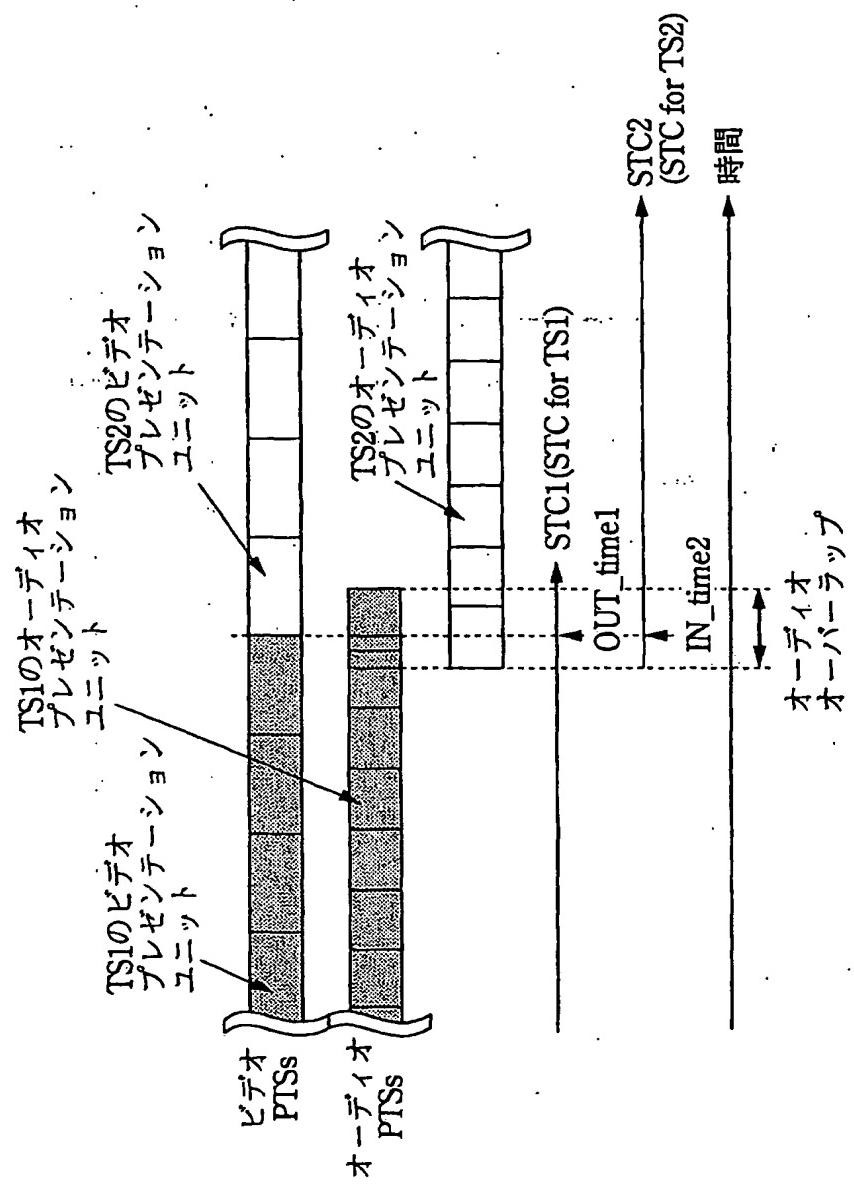


FIG.93

86/93

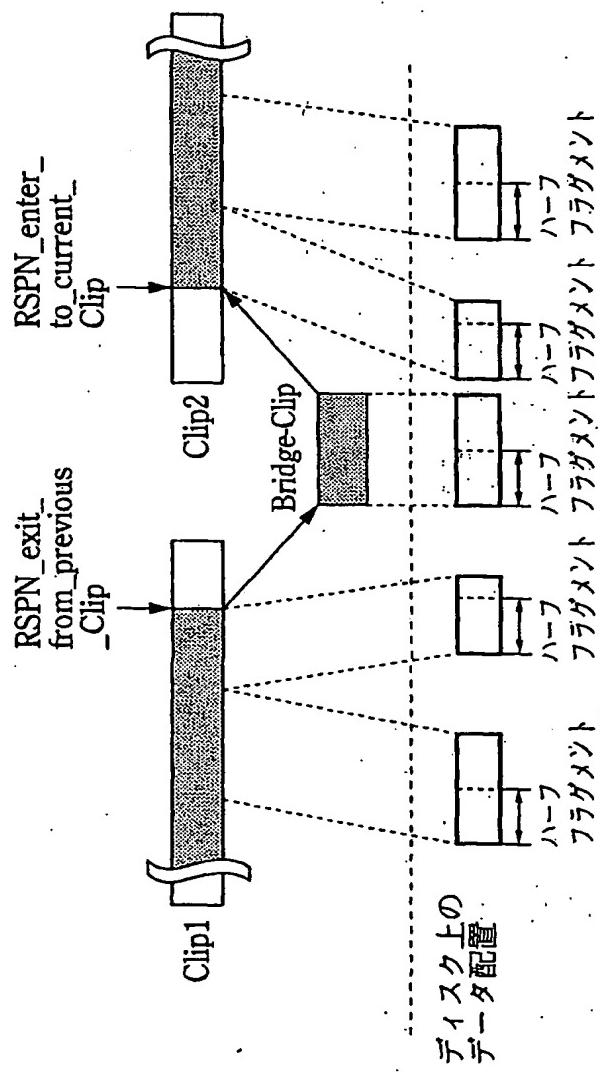


FIG.94

87/93

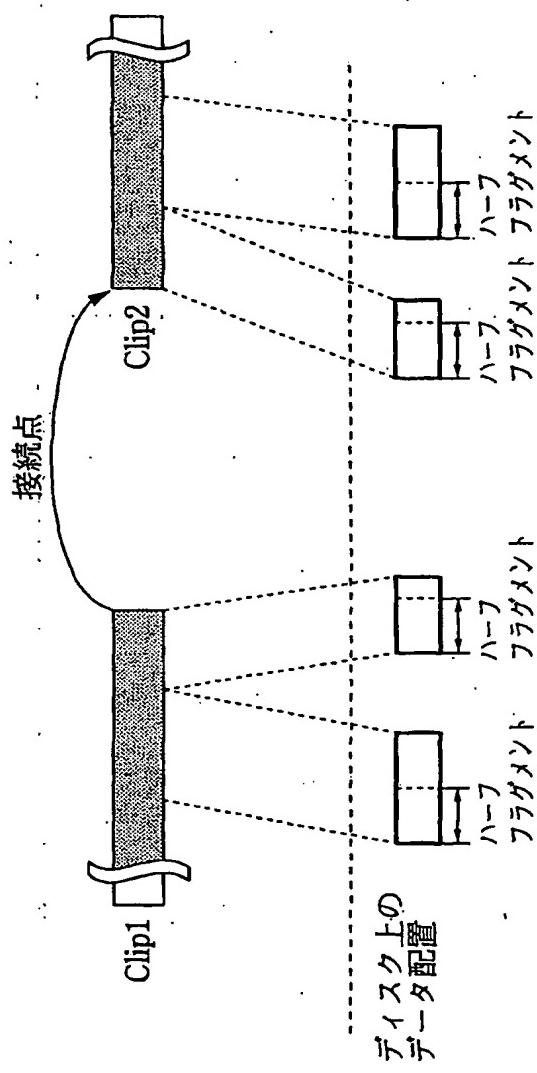


FIG.95

88/93

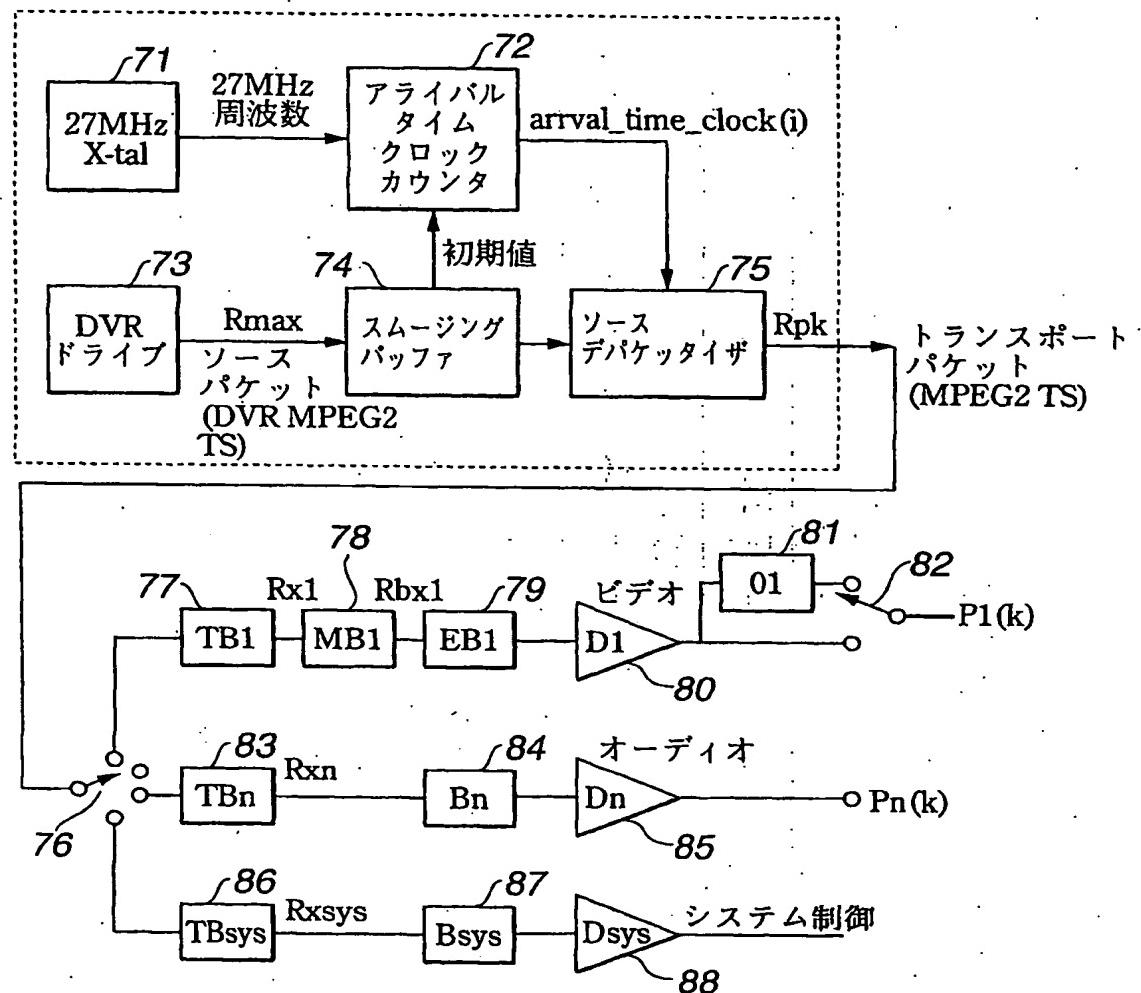


FIG.96

89/93

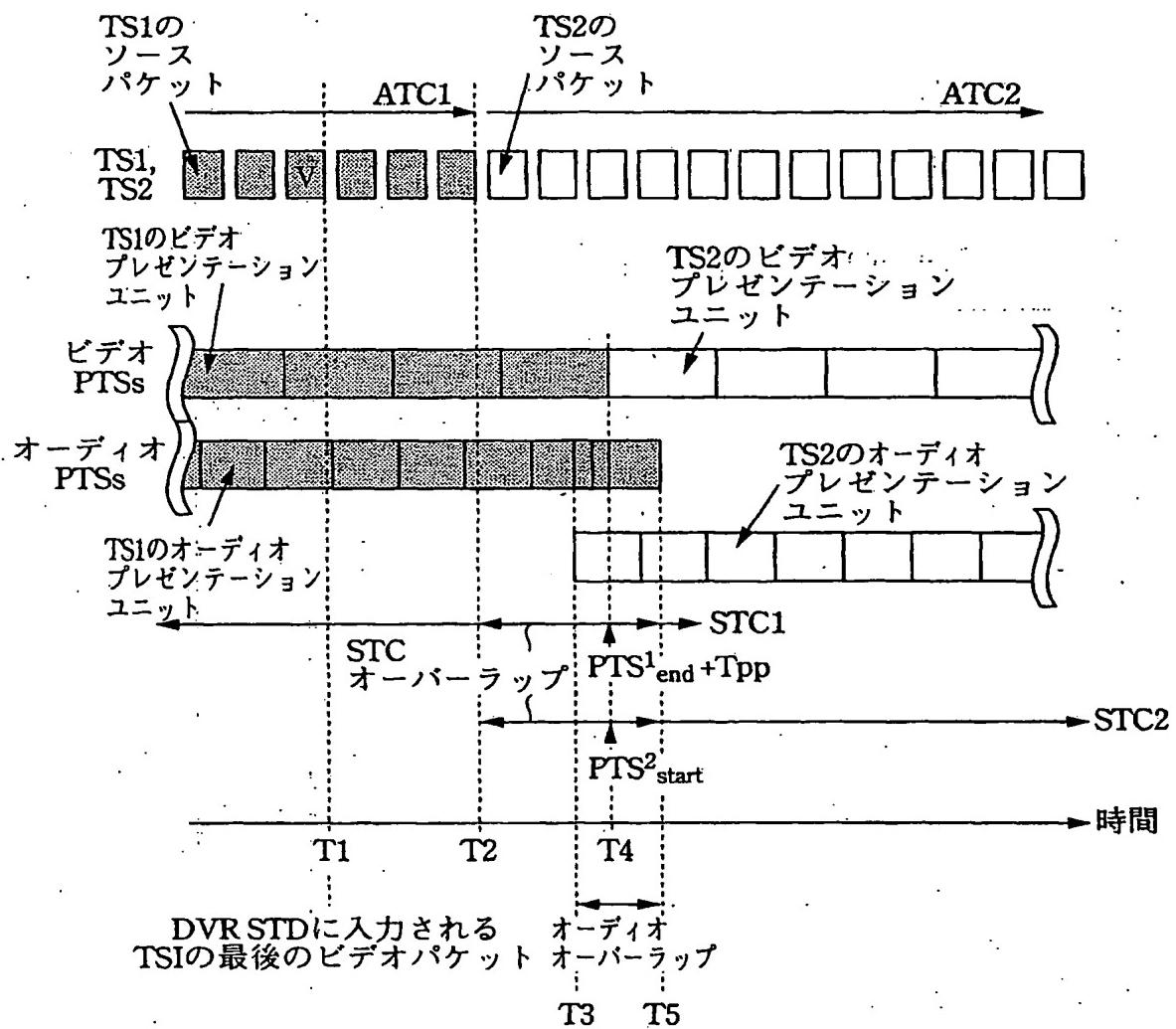


FIG.97

90/93

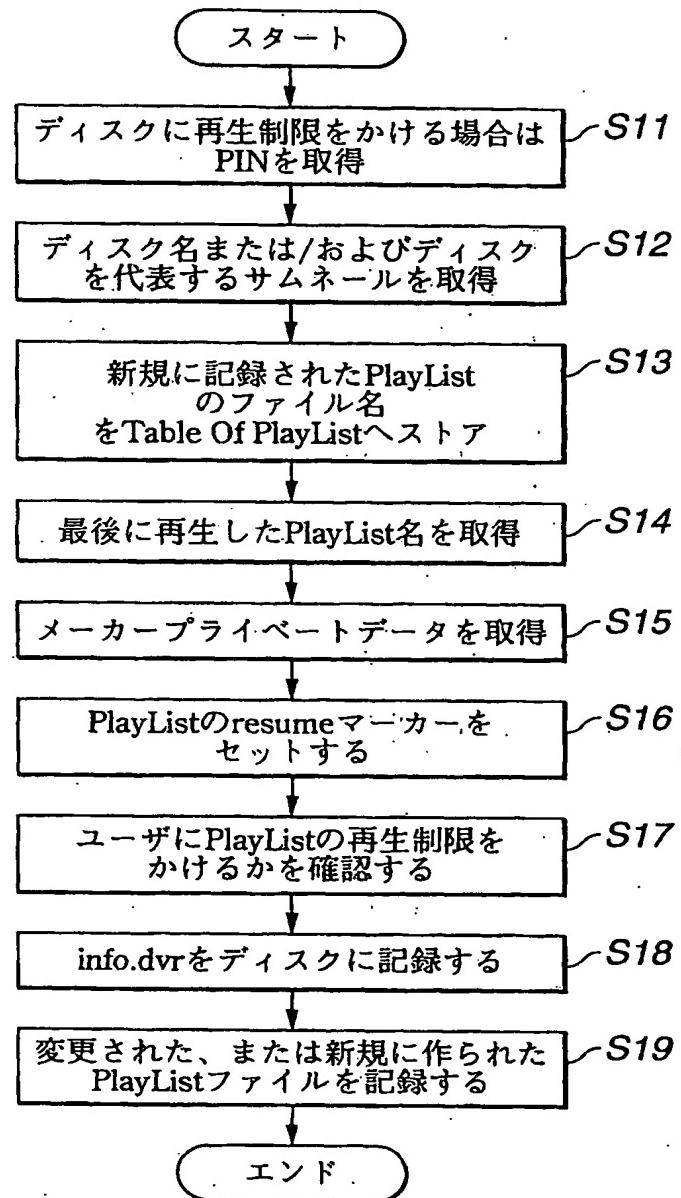


FIG.98

91/93

**FIG.99**

92/93

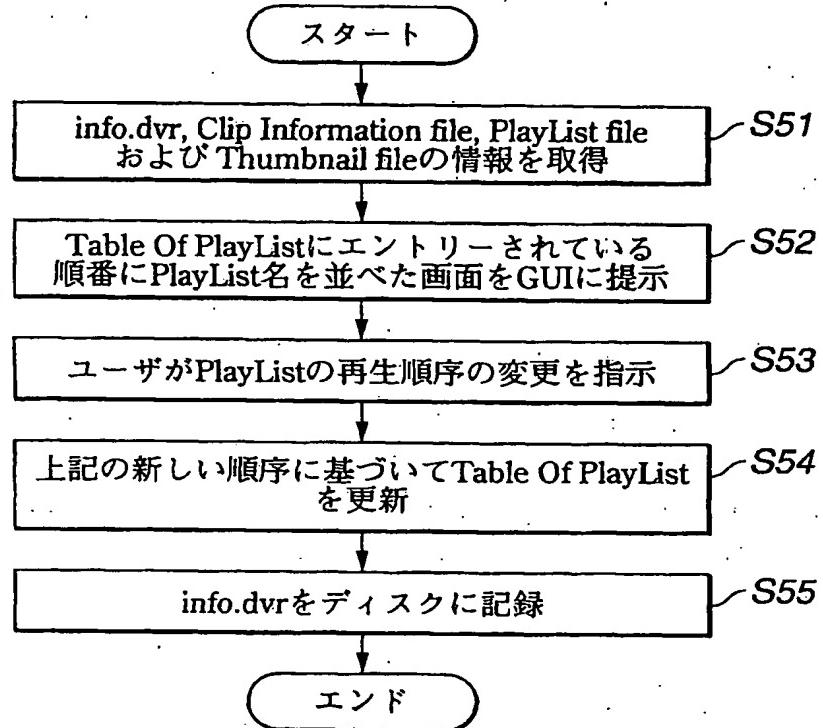


FIG.100

93/93

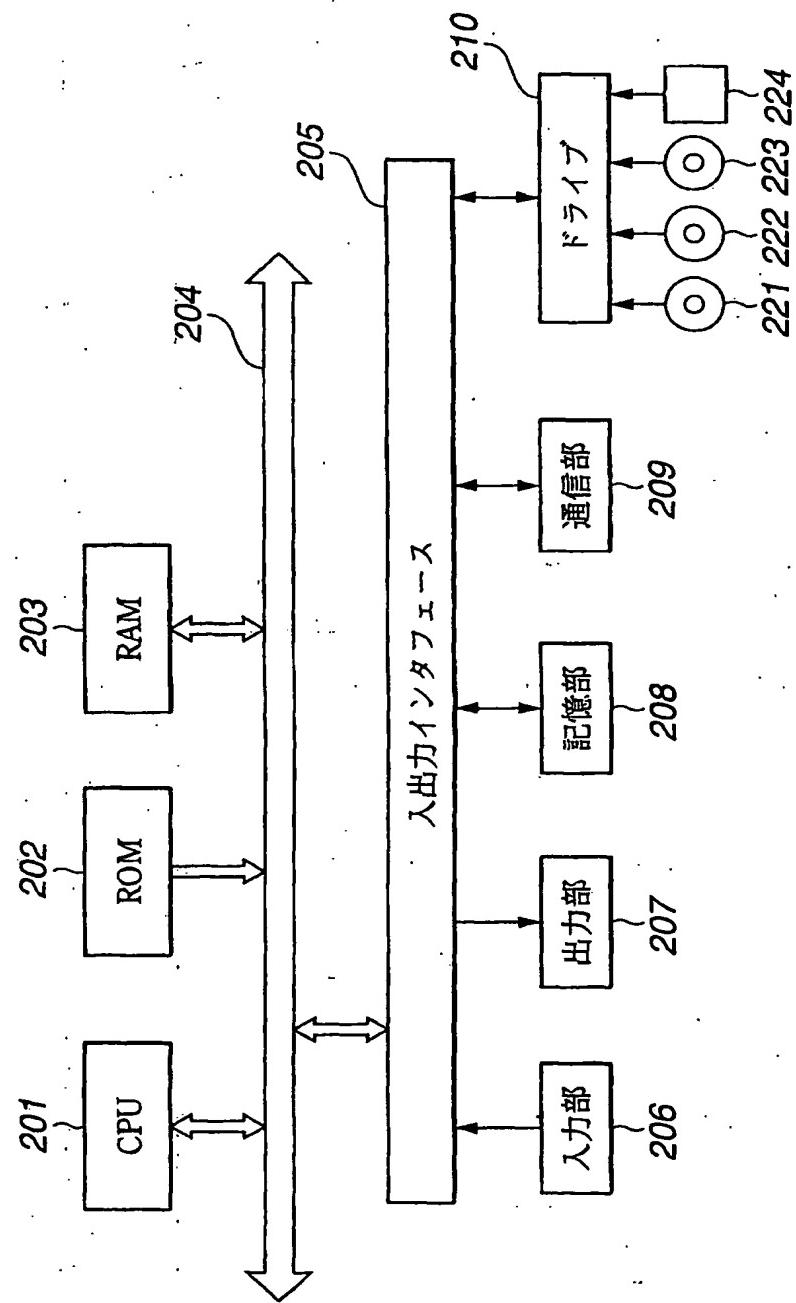


FIG.101

INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP01/03418

A. CLASSIFICATION OF SUBJECT MATTER
Int.Cl⁷ H04N 5/93, G11B 20/10

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

Int.Cl⁷ H04N 5/76-5/956, G11B 20/10-20/12Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
Jitsuyo Shinan Koho 1922-1996 Toroku Jitsuyo Shinan Koho 1994-2001
Kokai Jitsuyo Shinan Koho 1971-2001 Jitsuyo Shinan Toroku Koho 1996-2001

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	JP, 09-106631, A (Sony Corporation), 22 April, 1997 (22.04.97), Full text; Figs. 1 to 35 & AU, 6630496, A & CA, 2200346, A & EP, 784848, A & BR, 9606567, A & CN, 1213832, A	1-23
Y	JP, 2000-50197, A (Victor Company of Japan, Limited), 18 February, 2000 (18.02.00), Full text; Figs. 1 to 11 & EP, 1018743, A & CN, 1259737, A	1-23
Y	JP, 10-092155, A (Toshiba Corporation), 10 April, 1998 (10.04.98), Full text; Figs. 1 to 22 (Family: none)	10-14

 Further documents are listed in the continuation of Box C. See patent family annex.

- * Special categories of cited documents:
 "A" document defining the general state of the art which is not considered to be of particular relevance
 "E" earlier document but published on or after the international filing date
 "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
 "O" document referring to an oral disclosure, use, exhibition or other means
 "P" document published prior to the international filing date but later than the priority date claimed

- "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
 "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
 "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
 "&" document member of the same patent family

Date of the actual completion of the international search
12 June, 2001 (12.06.01)Date of mailing of the international search report
19 June, 2001 (19.06.01)Name and mailing address of the ISA/
Japanese Patent Office

Authorized officer

Facsimile No.

Telephone No.

国際調査報告

国際出願番号 PCT/JP01/03418

A. 発明の属する分野の分類(国際特許分類(IPC))

Int Cl' H04N 5/93, G11B 20/10

B. 調査を行った分野

調査を行った最小限資料(国際特許分類(IPC))

Int Cl' H04N 5/76-5/956, G11B 20/10-20/12

最小限資料以外の資料で調査を行った分野に含まれるもの

日本国実用新案公報	1922-1996年
日本国公開実用新案公報	1971-2001年
日本国登録実用新案公報	1994-2001年
日本国実用新案登録公報	1996-2001年

国際調査で使用した電子データベース(データベースの名称、調査に使用した用語)

C. 関連すると認められる文献

引用文献の カテゴリー*	引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示	関連する 請求の範囲の番号
Y	JP, 09-106631, A (ソニー株式会社) 22. 4月. 1997 (22. 04. 97) 全文、第1-35図 & AU, 6630496, A & CA, 2200346, A & EP, 784848, A & BR, 9606567, A & CN, 1213832, A	1-23
Y	JP, 2000-50197, A (日本ビクター株式会社) 18. 2月. 2000 (18. 02. 00) 全文、第1-11図 & EP, 1018743, A & CN, 1259737, A	1-23
Y	JP, 10-092155, A (株式会社東芝) 10. 4月. 1998 (10. 04. 98) 全文、第1-22図 (ファミリーなし)	10-14

 C欄の続きにも文献が列挙されている。 パテントファミリーに関する別紙を参照。

* 引用文献のカテゴリー

「A」特に関連のある文献ではなく、一般的技術水準を示すもの

「E」国際出願日前の出願または特許であるが、国際出願日以後に公表されたもの

「L」優先権主張に疑義を提起する文献又は他の文献の発行日若しくは他の特別な理由を確立するために引用する文献(理由を付す)

「O」口頭による開示、使用、展示等に旨及する文献

「P」国際出願日前で、かつ優先権の主張の基礎となる出願

の日の後に公表された文献

「T」国際出願日又は優先日後に公表された文献であって出願と矛盾するものではなく、発明の原理又は理論の理解のために引用するもの

「X」特に関連のある文献であって、当該文献のみで発明の新規性又は進歩性がないと考えられるもの

「Y」特に関連のある文献であって、当該文献と他の1以上の文献との、当業者にとって自明である組合せによって進歩性がないと考えられるもの

「&」同一パテントファミリー文献

国際調査を完了した日

12. 06. 01

国際調査報告の発送日

19.06.01

国際調査機関の名称及びあて先

日本国特許庁 (ISA/JP)

郵便番号 100-8915

東京都千代田区霞が関三丁目4番3号

特許庁審査官(権限のある職員)

鈴木 明

印

5C 9185

電話番号 03-3581-1101 内線 3541